Commodore INFO

PRIJS ƒ 7,95 / Bfr. 160



ONAFHANKELIJK BLAD VOOR AMIGA- EN C64- GEBRUIKERS Jaargang 7, NO.7, okt./nov. '90

8-UP!

AMOS getest

GEOS INFO

Nieuwe spellen op de Amiga

Beeldmanipulaties op de C64

Computer Info Beurs 3 november RAI Amsterdam

COLOFON

Commodore INFO is een uitgave van:
Sala Communications
Postbus 43048
1009 ZA Amsterdam

Uitgever:
V.Sharfman

Redactie:
ir. L. Sala (hoofdredacteur),
W.A. Scheer (eindredacteur),
drs. U. Schuurmans,
R.Goudriaan, B. Venema,
P. Boncz, MGCC/Johan&Johan,
Michel de Boer, Hylke Sprangers,
telefoon: 020-228871

Produktie:
drs. H. Zoete,
J. Broekhuizen

Advertentie-exploitatie:
ing. V. Sala, D. van Vlijmen
ing. B. Sala, H. Bia
telefoon: 020-273198

Abonnementen en administratie:
Marjo Jansen
telefoon: 020-248006

Vragen betreffende abonnementen ontvangen wij bij voorkeur schriftelijk, met meesturen van het omslagetiket.
Telefonisch uitsluitend tussen 12.00 en 15.00 uur.

Vragen over artikelen:
Voor vragen en opmerkingen over artikelen of anderszins graag een briefje aan de redactie (zie boven).

Listingtelefoon:
02155-25162
(ma: 17.00-21.00 u)

Geos Gebruikers Groep:
Postbus 52,
1300 AB Almere
03240-10041

Cover:
Sierra Online

Zetwerk & druk:
NDB, Zoeterwoude

Distributie:
In Nederland: Betapress, Gilze
In België: AMP, Brussel

© 1990 Commodore INFO
Alle rechten voorbehouden
ISSN: 0169-3085



Inhoud



Redactioneel

Efficiency Beurs 1990. Commodore Nederland maakt bekend dat, zoals verwacht, zij zich definitief zal gaan storten op multi-media. De auteurstaal AmigaVision wordt daartoe de standaard en vanaf Amiga 2000 meegeleverd.
Tijdens de EB is nogmaals duidelijk geworden dat Commodore van haar spelletjes-imago af wil; het bedrijf wil op de professionele toer. De 64/128 had al afgedaan, nu is het lot schoren aan de Amiga 500. De aandacht gaat vooral naar de 3000. Over een 2000 werd ook nauwelijks meer een woord gerept.
De vraag die dan ook direct op ons af komt is of deze professionele firma niet honderdduizenden enthousiaste gebruikers in de steek laat. Zij laten immers steeds weer zien, dat er uit de hobby-computers het onmogelijke kan worden gehaald.
Gelukkig hoeven die gebruikers niet persé op Commodore Nederland terug te vallen. Zolang er nog bedrijven zijn als KCS uit Dordrecht, kunnen ze profiteren van schitterende uitvinden; met name voor deze typen computers.
Aan hobby-isten ontbreekt het niet. Steeds meer 64-gebruikers raken in de ban van GEOS. Getuige de grote aantallen brieven van lezers met vragen en opmerkingen over dit programma. En toegegeven: GEOS wordt steeds professioneler. Voor de GEOS-fans hebben we dan ook weer de nodige informatie.
Voor dit nummer bekeek de redactie AMOS-Basic op de Amiga. Een hele verbetering ten opzichte van het (oude) Amiga-Basic. Aan spellen ontbreekt het deze keer ook niet. Van Sierra en Microprose het nieuwste van het nieuwste.
Ook de fabrikanten van al dit moois bewijzen daarmee dat de onderste, nu onderschatte lijn van Commodore nog volop in beweging is.
De vraag is alleen hoelang derden nog stand kunnen houden als de maker niet meer achter zijn creatie staat.

Wilbert Scheer

Vaste rubrieken:

Listings (C64):

Excuses...

We hadden in dit nummer een test willen opnemen over het PC Powerboard van KCS. Maar gezien de updates die KCS binnenkort hoopt uit te brengen, heeft de redactie besloten nog even te wachten met het publiceren hiervan. Nog even geduld alstublieft.

De redactie

Abonnement:

Tot het einde van het jaar (1 nummer): *f* 7,50 of Brf. 120 Betaling op giro 4985259 (België: BBL nr. 310050602562) t.n.v. SAC/Commodore-Info. Oude nummers kunt u alleen krijgen bij vooruitbetaling van *f* 6,75 op de bovenstaande rekening.
Ook telefonisch bestellen is mogelijk. Bel GRATIS 06-02242222 (teleservice) elke dag van 20.20 uur (dus ook in het weekend). België: 115555, dagelijks tot 22.00 uur. Deze nummers zijn allen bedoeld voor de opgave van nieuwe abonnementen.
Opzegging dient schriftelijk te geschieden uiterlijk twee maanden voor de aanvang van een nieuwe abonnementsperiode van een jaar.



GRAPHICS OP DE 64

Deel 12: Fractals

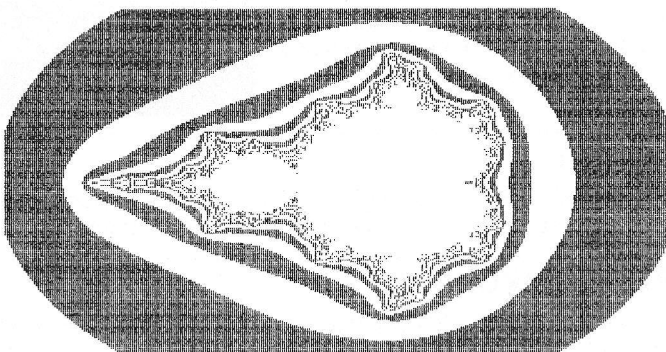
Zoals elke cursus heeft ook deze cursus een einde. Dat einde is nu helaas gekomen. In deze laatste aflevering zullen we verder gaan met het boeiende onderwerp fractals. Aan het eind van deze aflevering zullen we ook nog een listing geven van een programma dat een grote gekleurde bal over het scherm doet stuiteren. Dit programma toont dat er met goed programmeerwerk zeker spectaculaire grafische effecten uit de Commodore 64 kunnen worden gehaald.

Tevens zullen we in deze aflevering nog een overzicht geven van alle adressen van de VIC-chip. Wat betreft de fractals gaan we nu over tot het echte werk, dat u waarschijnlijk wel kent van de vele kleurige plaatjes. We behandelen onder andere de Mandelbrot-set en de Julia-fractals. Hoewel deze fractals gigantisch veel rekenwerk kosten (dit loopt al gauw in de uren), zijn er zeker zeer aardige resultaten te bereiken op de Commodore 64. Kijk bijvoorbeeld alvast maar naar figuur 1, voor een fraai 64-plaatje van de Mandelbrot-set. We wensen u veel plezier met deze laatste aflevering.

Mandelbrot-set

De beroemdste fractal, en ook zeker een van de mooiste, is ongetwijfeld de fractal van Mandelbrot. Deze fractal, die in figuur 1 te bewonderen is, is ontdekt door Benoit Mandelbrot, een van oorsprong Poolse wiskundige. Deze Mandelbrot mag zonder twijfel de Vader van de Fractals worden genoemd. Het karakteristieke figuurtje in de fractal van Mandelbrot wordt ook wel het appelmannetje (of Mandelbrot-set) genoemd. De theorie achter deze fractal is echter nogal ingewikkeld en vindt haar oorsprong in de complexe getaltheorie. Complexe getallen gaan nog een stapje verder dan de reële getallen en zijn alle van de vorm a+bi. Daarbij zijn a en b reële getallen, en is i de wortel uit -1. We vinden het hier echter veel te ver gaan om de hele theorie van de complexe getallen uit te leggen. Omdat we toch een klein beetje willen vertellen hoe de fractal van Mandelbrot tot stand komt, leggen we dit hieronder in versimpelde vorm uit. Hierbij zullen we ons niet bezig houden met complexe getallen. Het grafische scherm bestaat, zoals u weet, uit 64000 pixels, 320 horizontaal en 200 verticaal. Elk pixel geven we weer een x- en een y-coördinaat. Daarbij heeft het pixel linksboven coördinaten (0, 0) en het pixel rechtsonder (319, 199). Elk pixel heeft dus bepaalde (x, y) coördinaten. We bekijken nu de volgende formule: r2 = r*r - i*i + x i2 = 2*r*i + y

Voor elk pixel op het scherm met coördinaten (x, y) gaan we bovenstaande formule evalueren (i.e. invullen). Daarbij stellen we in het begin de variabelen r en i op 0. Als we de formule in hebben gevuld, heeft r2 de waarde x, en i2 de waarde y, want de variabelen r en i zijn (nog) gelijk aan 0. Bedenk hierbij dat x en y de coördinaten van het gekozen pixel zijn. Vervolgens geven we r de waarde van r2 en i de waarde van i2 en evalueren we de formule opnieuw. De x en y veranderen hierbij niet! We krijgen dan een nieuw resultaat voor r2 en i2. Nu kunnen we r en i weer gelijk maken aan r2 en i2 en de formule nogmaals evalueren. Dit proces kunnen we zo steeds herhalen. We vullen als het ware de uitkomst van de formule steeds opnieuw in de formule in. Als we dit een keer of veertig herhalen blijkt dat de waarde van het paar (r2, i2) of naar oneindig gaat (aangetrokken door oneindig), of naar nul nadert. We zeggen nu dat het pixel met de coördinaten (x, y) of naar oneindig gaat, of naar nul gaat. Dit proces werd door Mandelbrot ontdekt, en dient als basis voor zijn fractals. We vatten eerst het bovenstaande nog even samen. Voor elk pixel van het scherm met coördinaten (x, y) wordt de bovenstaande formule ingevuld. De uitkomst wordt daarna steeds weer in de formule zelf ingevuld, waarbij alleen de x en y steeds gelijk blijven. Als we een stuk of veertig keer deze formule hebben geëvalueerd, blijkt dat het paar (r2, i2) of rond de nul schommelt, of naar oneindig gaat. Aan de hand hiervan kunnen we de pixels aan of uit zetten. We kunnen bijvoorbeeld alle pixels die naar oneindig gaan, aan zetten. Wat we dan krijgen is de omtrek van het appelmannetje. In **figuur 1** kunt u echter ook nog allemaal banen om het appelmannetje heen zien. Dit komt omdat sommige punten sneller naar oneindig gaan dan andere. Door de pixels in groepen te verdelen, die variëren in de snelheid waarin ze naar oneindig gaan, worden de mooie banden verkregen. Daarbij zijn alle pixels



Figuur 1



die naar 0 gaan, wit gelaten (uit). Het programma voor de Mandelbrot-fractal volgt nu:

```
5 rem mandelbrot
10 hires 0,1
15 xb=-2.5:xe=1.5
16 yb=-1.5:ye=1.5
20 fory=0to199:b=y/200*(ye-yb)+yb
30 forx=0to319:a=x/320*(xe-xb)+xb
40 r=0:i=0:t=0
50 z=r:r=r*r-i*i+a:i=2*z*i+b
60 ifr*r+i*i>16then plot x,y,-(int(t/2)*2=t):goto80
70 t=t+1:ift<40(then50
80 next:next
90 goto 90
```

Het programma is in Simon's Basic geschreven. In de regels 20 en 30 worden er 2 loops gedefinieerd die alle pixels op het scherm langs lopen. Omdat de Mandelbrot-fractal heel klein om de het punt (0,0) zit (van -2.5 tot 1.5 op de x-as en van -1.5 tot 1.5 op de y-as), moet elk pixel nieuwe coördinaten krijgen. Het pixel linksboven moet coördinaten (-2.5,-1.5) krijgen en het pixel rechtsonder (1.5,1.5). In regels 15 en 16 staan de waarden van deze coördinaten. (xb,yb) is het linker bovenpixel en (xe,ye) het rechter onder. De andere pixels krijgen coördinaten die hier tussenin liggen. Deze coördinaten worden in regels 20 en 30 berekend. (a,b) zijn de nieuwe coordinaten van een pixel. Deze coördinaten worden gebruikt bij het evalueren van de eerder besproken formule. In regel 50 staat de eigenlijke formule. In regel 60 wordt gekeken of het pixel naar oneindig gaat. Als dit zo is wordt het pixel aangezet, mits het in een band zit die 'aan' is. Omdat voor elk pixel de formule een heleboel keer doorlopen moet worden, kost het tekenen van zo'n fractal gigantisch veel tijd. Het bovenstaande programma doet er bijvoorbeeld 7 uur en 57 minuten over voordat de fractal helemaal af is. De Mandelbrot-fractal is echter symetrisch in de x-as. Oftewel, we kunnen elk punt spiegelen in de x-as. Dit levert een tijd-besparing van bijna 50% op. Het tekenen van de fractal kost dan nog 'slechts' 4 uur. U moet dan in regel 60 na de 'then' nog het volgende toevoegen:

```
plot x,200-y,-(int(t/2)*2=t):
```

De loop in regel 20 hoeft dan nog maar tot de 100 te lopen. U zult zich misschien afvragen hoe Mandelbrot aan zo'n 'vreemde' formule komt. Wel, als de formule uitgeschreven wordt met complexe getallen, komt er een hele simpele formule te staan:

z2 = z*z + c

In de bovenstaande formule zijn z en c complexe getallen. U ziet dat de formule dan ineens heel simpel is. In de vorige aflevering hebben we verteld dat fractals zich kenmerken door een bepaalde zelfgelijkvormigheid. Dus dat er bepaalde motieven steeds terugkomen bij uitvergroting van de figuur. Het zelfgelijkvormigheid-motief bij de Mandelbrot-fractal is, hoe kan het ook anders, het appelmannetje. Bij uitvergroting van een stuk rand van dit appelmannetje, komen er steeds weer appelmannetjes terug. Dit uitvergroten kunt u ook met het bovenstaande programma doen, door de coördinaten in regels 15 en 16 te veranderen. We kunnen een klein rechthoekje uit het plaatje vergroten door xb,yb gelijk te maken aan de coördinaten van de linker bovenhoek van dit rechthoekje en xe,ye aan de coördinaten van de rechter onderhoek. Als voorbeeld gaan we het kwart linksboven vergroten. De coördinaten van de linker bovenhoek van dit kwart zijn (-2.5,-1.5) en van de rechter onderhoek (-.5,0) (dit is het midden van het appelmannetje). Regels 15 en 16 moet u dan als volgt veranderen:

```
15 xb=-2.5:xe=-.5 16 yb=-1.5:ye=0
```

Het plaatje is nu niet meer symetrisch en we kunnen dus geen gebruik maken van de eerder genoemde versnelling! Bij uitvergroting van de rand van het appelmannetje moet de formule meer dan 40 keer geëvalueerd worden. U moet dan de waarde in regel 70 aanpassen.

Julia-fractals

De tweede soort fractals die we bespreken in deze aflevering, zijn de Julia-fractals. Deze soort fractals zijn ook zeer bekend, en minstens net zo mooi als fractals van de Mandelbrot-set. Bij de Julia-fractals gaan we weer van dezelfde formule uit als bij de Mandelbrot-fractal:

x2 = x*x - y*y + a y2 = 2*x*y + b

In de bovenstaande formule hebben we de variabelen andere namen gegeven. Als u echter goed kijkt, zult u zien dat deze formule exact gelijk is aan de formule die we bij de Mandelbrot-fractal hebben gegeven. Waarom we andere namen aan de variabelen hebben gegeven, zullen we weldra zien. We gaan weer voor elk pixel op het scherm de bovenstaande formule evalueren. We nemen weer telkens een pixel, met coördinaten (x, y). De formule wordt weer ingevuld, met daarbij voor a en b goedgekozen constanten. U ziet dus dat de pixel-coördinaten ergens anders in de formule worden ingevuld, dan dat bij de Mandelbrot-fractal het geval was. Ook nu moet de uitkomst van de formule steeds weer in de formule zelf ingevuld worden. Hierbij blijven a en b steeds constant. Straks zullen we een aantal mooie waarden voor a en b bekijken. Een pixel kan weer naar oneindig, of naar nul gaan. Aan de hand daarvan kleuren we de pixel met zwart of wit. Hier volgt het programma voor het tekenen van Julia-fractals:

```
5 rem julia-fractals
10 hires 0,1:a=-.75:b=0
20 fory=0to199
30 forx=0to319:r=x*4/320-2
40 i=y*3/200-1.5:t=0
50 z=r:r=r*r-i*i+a:i=2*z*i+b
60 ifr*r+i*i>25then plot x,y,-(int(t/2)*2=t):goto80
70 t=t+1:ift<40then50
80 next:next
90 goto 90
```

Ook dit programma is weer in Simon's Basic geschreven, en lijkt veel op het programma voor de Mandelbrot-fractal. In het algemeen blijkt dat het tekenen van Julia's nog een stuk langer duurt dan het tekenen van de Mandelbrot-fractal. U kunt dan ook het best uw computer een nachtje laten aanstaan, als u met Julia-fractals gaat experimenteren. In regel 10 worden de constanten a en b gedefinieerd. Als voor (a, b) de waarden (-.75, 0) worden gekozen, wordt de beroemde San-Marco fractal verkregen. In de tabel in **figuur 2** staat nog een aantal leuke waarden die u kunt uitproberen. Overigens kunt u ook bij dit programma de wachttijd verkorten door het geheel in de x-as te spiegelen. De aanpassingen zijn hetzelfde als bij het Mandelbrot programma. We hebben steeds gezegd dat een pixel naar oneindig, of naar nul kan gaan (bij evaluatie van de formule). Feitelijk is er ook nog een derde optie die we niet hebben genoemd. In de bovenstaande programma's hebben we deze moge-

Julia Fractals	
a	b
0.75	0
-0.75	0
0	1
0.8	0.4
-0.6	-0.2
0.25	0.5

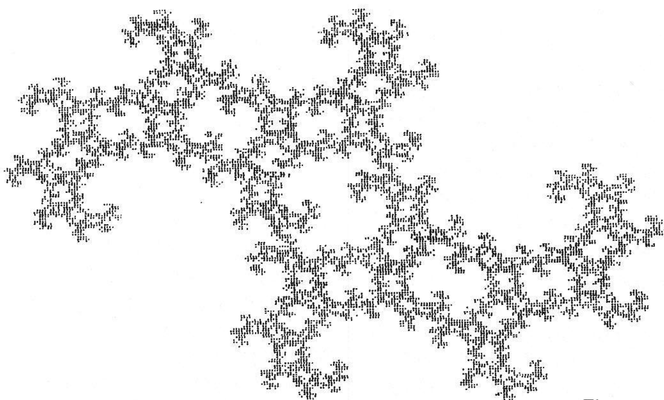
Figuur 2



lijkheid, uit snelheids- en complexiteits-oogpunt, ook steeds weggelaten. In de besproken fractals bestaan er zogeheten aantrekkende gebieden. Als derde mogelijkheid kan een pixel ook aangetrokken worden door zo'n gebied. Door de pixels die worden aangetrokken door zo'n gebied ook nog verschillend te kleuren, ontstaan er de mooiste plaatjes. Voor de Commodore 64 gaat dit helaas iets te ver: de rekentijd kan dan in dagen worden uitgeteld.

Zandfractals

Als laatste uit de grote familie der fractals willen we u laten kennismaken met de zandfractals. Dit zijn fractals die opgebouwd zijn uit losse punten. Deze fractals danken hun naam aan de manier waarop zij ontstaan. Op het scherm verschijnen allemaal losse puntjes op schijnbaar willekeurige plaatsen. Na verloop van tijd blijkt er echter toch een bepaald verband tussen de punten te bestaan en ontstaat er een schitterend beeld. Het lijkt dus alsof er iemand zandkorrels uitstrooit over een vel wit papier. Niet alleen het eindresultaat maar ook het ontstaan van een zandfractal is fascinerend om te zien. In **figuur 3** kunt u een zeer fraaie zandfractal bewonderen. De zelfgelijkvormigheid is in deze fractal overduidelijk aanwezig. De zelfgelijkvormigheid van zandfractals ontstaat door het gebruik van zogenaamde gelijkvormigheidstransformaties. Een transformatie is een operatie die kan worden uitgevoerd op elk punt van een figuur, waardoor de coördinaten van elk punt worden omgezet in nieuwe coördinaten. Een eenvoudige operatie is bijvoorbeeld het naar rechts schuiven van een punt. Dit kan gedaan worden door de x-coördinaat van het punt met 1 te verhogen. Als we dit voor elk punt van een willekeurig figuur doen, dan is het totale effect van de transformatie een verschuiving naar rechts van de gehele figuur. Een gelijkvormigheidstransformatie is een transformatie waarbij de vorm van het figuur onveranderd blijft door de transformatie. Let wel, we spreken hier alleen over de vorm en niet over de grootte van de figuur. De afmetingen van de figuur mogen bij de transformatie rustig veranderen; het is voor fractals zelfs essentieel dat deze veranderen. Hieronder zullen we drie belangrijke gelijkvormigheidstransformaties bespreken.



Figuur 3

1. Rotatie

Een rotatie draait een figuur om een bepaald punt (draaipunt). De grootte van de figuur verandert niet door een rotatie. De rotatie wordt geheel bepaald door de grootte van de draaihoek en de coördinaten van het punt. Een positieve draaihoek betekent dat de figuur tegen de wijzers van de klok in wordt gedraaid. De wiskundige beschrijving van een rotatie is als volgt:

x2 = (x-mx)*cos(a) - (y-my)*sin(a) + mx y2 = (x-mx)*sin(a) + (y-my)*cos(a) + my

Hierbij zijn (mx,my) de coördinaten van het draaipunt en is a de grootte van de draaihoek. Om nu een bepaald punt te roteren vullen we voor x en y de coördinaten van dat punt in. Na berekenen bevatten x2 en y2 de nieuwe coördinaten.

2. Schaalverandering

Bij een schaalverandering worden de afstanden tussen een gekozen punt (het schalingspunt) en de punten van een figuur met een bepaalde factor vermenigvuldigd. Afhankelijk van de factor wordt de figuur dan groter of kleiner en komt de figuur verder van of juist dichter bij het schalingspunt te liggen. De wiskundige beschrijving is als volgt:

x2 = f*(x-mx) + mx y2 = f*(y-my) + my

Hierbij zijn (mx,my) de coördinaten van het schalingspunt en is f de schalingsfactor. Voor het uitvoeren van de transformatie moeten voor x en y weer de coördinaten van een punt worden ingevuld.

3. Schaalrotatie

Dit is eigenlijk geen echt nieuwe transformatie. We noemen hem hier echter toch apart, omdat de schaalrotatie een belangrijke transformatie is voor het verkrijgen van mooie fractals. De schaalrotatie krijgen we door het achterelkaar toepassen van een rotatie en een schaalverandering. Daarbij moeten het draaipunt van de rotatie en het schalingspunt van de schalingsverandering hetzelfde punt zijn.

Met deze gelijkvormigheidstransformaties kunnen we nu op eenvoudige wijze fractals construeren. We kiezen eerst een aantal verschillende transformaties uit; bijvoorbeeld twee schaalrotaties. Vervolgens kiezen we een willekeurig startpunt. Op dit punt passen we een van de twee transformaties toe. De keuze van de transformatie laten we aan het toeval over. Het zo verkregen nieuwe punt zetten we aan. Vervolgens passen we weer willekeurig een van de transformaties toe en zetten het nieuwe punt weer aan. De fractal wordt zo punt voor punt opgebouwd. Dit herhalen we totdat de fractal duidelijk genoeg is (bijvoorbeeld 20.000 punten). Hieronder volgt een voorbeeld programma:

```
10 rem zandfractal
20 hires 0,15
30 x=160:y=100
40 q=rnd(-ti)
50 plot x,y,1
60 a=int(rnd(0)*2):on a+1 gosub 100,150
70 goto 50
95 rem schaalrotatie: hoek = {pi}/4.5
96 rem draaipunt = (160,150)
97 rem schalingsfactor = .667
100 x2=(x-160)*cos({pi}/4.5)-(y-150)*sin({pi}/4.5)+150
110 y2=(x-160)*sin({pi}/4.5)+(y-150)*cos({pi}/4.5)+150
120 x3=.667*(x2-160)+160
130 y3=.667*(y2-150)+150
140 x=x3:y=y3
145 return
147 rem schaalrotatie: hoek = {pi}/4.5
148 rem draaipunt = (160,50)
149 rem schalingsfactor = .667
150 x2=(x-160)*cos({pi}/4.5)-(y-50)*sin({pi}/4.5)+160
160 y2=(x-160)*sin({pi}/4.5)+(y-50)*cos({pi}/4.5)+50
170 x3=.667*(x2-160)+160
180 y3=.667*(y2-50)+50
190 x=x3:y=y3
200 return
```



```
1   rem *****************
5   rem * bouncing ball *
6   rem *****************
10  for i=6400 to 6855
20  read a:poke i,a:c=c+a:next
25  if c50503 then print"fout in data":end
30  x=16384
40  a=-5:y0=0:x0=0:vx=3:vy=43:t=0:vs=1
50  yc=y0+vy*t+.5*a*t*t:xc=xc+vx
60  if yc  then gosub 100
70  if xc then xc=0:vx=-vx
80  if xc295 then xc=295:vx=-vx
90  yc=200-yc:gosub200:printxc,yc:t=t+.4:goto
    50
100 yc=20:vy=vy-vs:y0=20:t=0
110 if vy then vs=-vs:ba=1
120 if ba=1 and vy=42 then 140
130 return
140 poke
    6768,x/256:poke6774,x-256*int(x/256):sys
    6678
200 by=8192+8*int(xc/8)+320*int(yc/8)+(yc
    and 7)
210 poke x,by-256*int(by/256):poke
    x+1,by/256
220 poke x+2,(xc and 7)/2:poke x+3,(yc and
    7):x=x+4:return
400 data 169,0,141,32,208,141,33,208
410 data 173,17,208,41,239,9,32,141
420 data 17,208,173,22,208,9,16,141
430 data 22,208,169,0,133,251,169,32
440 data 133,252,160,0,169,0,145,251
450 data 200,192,0,208,249,24,165,252
460 data 105,1,133,252,201,64,208,234
470 data 169,4,133,252,169,0,133,253
480 data 169,216,133,254,160,0,169,38
490 data 145,251,169,7,145,253,200,192
500 data 0,208,243,24,165,254,105,1
510 data 133,254,165,252,105,1,133,252
520 data 201,8,208,224,173,24,208,9
530 data 8,141,24,208,173,17,208,9
540 data 16,141,17,208,120,96,234,234
550 data 165,65,133,61,165,66,133,62
560 data 165,64,133,60,160,0,169,0
570 data 133,57,165,59,133,55,190,128
580 data 26,240,26,165,55,201,3,208
590 data 14,169,0,133,55,165,57,24
600 data 105,8,133,57,76,169,25,230
610 data 55,202,76,145,25,152,72,190
620 data 148,26,164,57,177,61,164,55
630 data 49,251,17,253,164,57,145,61
640 data 165,55,201,3,208,14,169,0
650 data 133,55,165,57,24,105,8,133
660 data 57,76,214,25,230,55,202,208
670 data 217,104,168,165,60,201,7,240
680 data 18,230,60,24,165,61,105,1
690 data 133,61,165,62,105,0,133,62
700 data 76,4,26,169,0,133,60,24
710 data 165,61,105,57,133,61,165,62
720 data 105,1,133,62,200,192,20,240
730 data 3,76,134,25,96,234,234,234
740 data 234,234,234,234,234,234,32,0
750 data 25,169,0,133,65,169,64,133
760 data 66,169,168,133,251,169,26,133
770 data 252,169,184,133,253,169,26,133
780 data 254,160,0,177,65,133,61,200
790 data 177,65,133,62,200,177,65,133
800 data 59,200,177,65,133,60,32,132
810 data 25,24,165,251,105,4,133,251
820 data 165,253,105,4,133,253,201,196
830 data 208,8,169,168,133,251,169,184
840 data 133,253,24,165,65,105,4,133
850 data 65,165,66,105,0,133,66,201
860 data 91,208,190,165,65,201,220,208
870 data 184,76,25,26,0,0,0,0
880 data 4,3,2,2,1,1,0,0
890 data 0,0,0,0,0,0,1,1
900 data 2,2,3,4,4,6,8,8
910 data 10,10,12,12,12,12,12,12
920 data 12,12,10,10,8,8,6,4
930 data 63,207,243,252,127,223,247,253
940 data 191,239,251,254,63,207,243,252
950 data 192,48,12,3,64,16,4,1
960 data 128,32,8,2,0,0,0
```

Het programma construeert een zandfractal m.b.v. twee schaalrotaties. In regel 30 worden x en y gelijk gemaakt aan de coördinaten van een startpunt. In regel 50 wordt het punt aangezet en in regel 60 wordt een van de twee transformaties gekozen. Het uitvoeren van de transformaties gebeurt in de subroutines op regels 100 en 150. De eerste transformatie is een schaalrotatie met als draaipunt (160,150), draaihoek pi/4.5 radialen (40 graden) en schalingsfactor 0.667. De tweede transformatie is een schaalrotatie met als draaipunt (160,50), draaihoek pi/4.5 radialen en schalingsfactor 0.667. Het programma is eenvoudig aanpasbaar. Vanaf regel 100 kunt u zelf subroutines plaatsen voor het uitvoeren van verschillende transformaties. In regel 60 zet u de startregels van de subroutines in de lijst achter de 'on a+1 gosub'-instructie. In het programma worden twee transformaties gebruikt. U kunt natuurlijk ook van meer transformaties gebruik maken. In regel 60 moet dan het getal 2 vervangen worden door een ander willekeurig getal.

Transformaties

Een andere nuttige transformatie voor het creëren van zandfractals is de spiegeling. We onderscheiden twee soorten spiegelingen: lijnspiegeling en puntspiegeling. De afmetingen van een figuur blijven bij spiegeling onveranderd. Hieronder geven we de wiskundige vergelijkingen voor deze transformaties.

Spiegelen in horizontale lijn: x2 = x y2 = my-y Hierbij is my de y-coördinaat van de lijn waarin gespiegeld wordt.

Spiegelen in verticale lijn: x2 = mx-x y2 = y Hierbij is mx de x-coördinaat van de lijn waarin gespiegeld wordt.

Spiegelen in een punt: x2 = mx-x y2 = my-y Hierbij zijn (mx,my) de coördinaten van het punt waarin gespiegeld wordt (het spiegelpunt).

Net zoals bij de rotatie kunnen vaak mooie fractals worden verkregen door de transformatie te combineren met een schalingsverandering. We moeten dan achter elkaar een puntspiegeling en een schalingsverandering uitvoeren, waarbij het spiegelpunt en het schalingspunt hetzelfde punt zijn. De transformatie heet dan -u raadt het al- een schaalspiegeling.

Bouncing Ball

Als laatste in deze aflevering, en ook als afsluiting van deze cursus, geven we nog een flitsend machinetaal programma. Het programma, Bouncing Ball, laat een bal, ter grootte van een sprite, steeds heen en weer op het scherm stuiteren. De bal zelf is echter geen sprite, maar wordt steeds opnieuw op het hires (het programma werkt dus in de hires-mode) getekend. Bij elke nieuwe positie wordt het balletje weer helemaal opnieuw op het hires getekend. Door de oude balletjes te laten staan, blijft de vloeiende baan van de bal op



het scherm zichtbaar. Het hires staat in de multi-color mode en de balletjes worden steeds om en om met rood, geel en blauw (op een zwarte achtergrond) gekleurd. De listing van dit programma staat aan het eind van dit artikel. In de regels 10 en 20 wordt het machinetaalprogramma in het geheugen gepoked (start sys 6678). In de volgende regels 30 tot 220 wordt de baan van het balletje berekend en in het geheugen gezet. Deze baan wordt van te voren berekend, opdat de uitvoer van de werkelijke routine dan nog sneller is. We hebben er nu een hele lange baan ingezet, zodat u eerst een minuut of vijf moet wachten voordat het werkelijke programma begint. Het machinetaalprogramma wordt daarna gestart met sys 6678. Als u wat aan de baan wilt veranderen, kunt u de parameters aanpassen vanaf regel 30. De baan die steeds wordt berekend ziet u door middel van (x, y) coördinaten op het scherm tijdens de berekening. U moet er voor uitkijken dat er geen negatieve coördinaten berekend worden: het machinetaalprogramma kan dan de mist ingaan, omdat het balletje dan buiten het scherm (midden in het geheugen) getekend wordt. U moet dan de parameters aanpassen.

Tot slot

We zijn nu echt aan het eind gekomen van de cursus 'Graphics op de 64'. In 12 afleveringen hebben wij u meegevoerd door de wereld van de graphics van de Commodore 64. Zo zijn we begonnen met het ontwerpen van eigen karaktersets. Vervolgens hebben we u kennis laten maken met de beweegbare sprites en zijn daarna aangeland bij het tekenen van prachtige high resolution schermen. Tenslotte hebben we in de laatste afleveringen enkele interessante onderwerpen op het gebied van de computerkunst belicht en heeft u zelf met deze onderwerpen kunnen experimenteren. Tot slot van deze cursus bieden wij u, zoals eerder beloofd, een overzicht van de hele video chip. In **figuur 4** staan alle in deze cursus behandelde adressen met een korte omschrijving. Dit overzicht kan zeer nuttig zijn bij het ontwikkelen van grafische programma's. We hopen dat u veel heeft opgestoken van de cursus en vooral veel plezier heeft gehad bij het experimenteren met graphics. Voor eventuele schriftelijke reacties kunt u zich wenden tot de redactie van de Commodore Info. Gegroet!

Michel de Boer &
Hylke Sprangers

Overzicht van de video chip

Adres	Functie omschrijving
53248	X-coordinaat sprite 0
53249	Y-coordinaat sprite 0
53250	X-coordinaat sprite 1
|	|
53263	Y-coordinaat sprite 7
53254	Most significant bits voor x-coordinaten van sprites
53265	bit 0-2: Verticale smooth-scrolling bit 3 : 24 of 25 regels tekst (0 = 24 regels, 1 = 25 regels) bit 4 : schakel scherm aan/uit (0 = uit, 1= aan) bit 5 : Keuze tekst/hires mode (0 = tekst, 1 = hires) bit 6 : Aanzetten extended color tekst mode (0 = uit, 1 = aan) bit 7 : Hi-bit raster scan lijn
53266	Huidige raster scan lijn
53269	Aanzetten van sprites
53270	bit 0-2: Horizontale smooth-scrolling bit 3 : 38 of 40 kolommen tekst (0 = 38 kolommen, 1= 40 kolommen) bit 4 : Aanzetten multicolor mode voor tekst en hires (0 = uit, 1 = aan)
53271	Vergroten van sprites in verticale richting
53272	bit 1-3: Pointer naar beginadres van een karakterset bit 3 : Pointer naar beginadres van hires (alleen in hires mode) bit 4-7: Pointer naar beginadres van het scherm
53275	Prioriteiten van sprites t.o.v. voorgrond graphics
53276	Aanzetten multicolor mode voor sprites
53277	Vegroten van sprites in horizontale richting
53278	Detecteren van botsingen tussen sprites onderling
53279	Detecteren van botsingen tussen sprites en voorgrond graphics
53280	Borderkleur
53281	Schermkleur
53282	Kleur van 01-bitpaar bij multicolor karakters en achtergrondkleur van karakters met ascii code 64-127 in extended color tekst mode
53283	Kleur van 10-bitpaar bij multicolor karakters en achtergrondkleur van karakters met ascii code 128-191 in extended color tekst mode
53284	Achtergrondkleur van karakters met ascii code 192-255 in extended color tekst mode
53285	Kleur van 01-bitpaar bij multicolor sprites
53286	Kleur van 11-bitpaar bij multicolor sprites
53287	Kleur van sprite 0
|	|
53294	Kleur van sprite 7

Figuur 4



BEELDMANIPULATIES (2)

De Rasterinterrupt

Zoals vorige keer gezegd, zullen we in deze aflevering het begrip rasterinterrupt bespreken. Dit onderwerp is van essentieel belang bij het maken van mooie vloeiende beeldmanipulaties. Met een rasterinterrupt kunnen we onder andere de hele opbouw van het scherm beïnvloeden. Om een goede rasterinterrupt te kunnen schrijven zijn een aantal adressen van belang. We zullen deze keer beginnen met het beschrijven van een aantal van deze adressen en daarna zullen we een voorbeeld geven van een rasterinterrupt en dit voorbeeld bespreken.

De adressen

Om een rasterinterrupt te kunnen programmeren zijn de volgende adressen van belang:

$D01A

IRQ Mask Register
Dit adres is nodig om de computer te vertellen op welke IRQ's gereageerd moet worden. Dit gebeurt aan de hand van een bepaald bitpatroon. Uit het volgende overzicht is op te maken welk bitpatroon nodig is bij welke toepassing.

bit 0 Als dit bit geset is vindt er een IRQ plaats als er een bepaalde rasterlijn bereikt is.
bit 1 Dit bit moet geset worden om de IRQ te laten plaatsvinden wanneer een sprite met de achtergrond in botsing komt. Dit wil dus zeggen: als de sprite bijvoorbeeld tegen een karakter aankomt, wordt er een IRQ gegeven.
bit 2 Dit bit zorgt er voor dat er gereageerd wordt op een sprite tegen sprite botsing.
bit 3 Dit bit wordt gebruikt bij lichtpen-toepassingen. Als het geset is geeft dat een IRQ indien er een signaal van de lichtpen komt.
bits 4-7 Deze bits worden niet gebruikt en je kunt ze dus invullen zoals je zelf wilt.

$D019

VIC Interrupt Flag Register
Als er IRQ heeft plaatsgevonden kan er aan de hand van $D019 bepaald worden om welke reden de IRQ is opgetreden. De bit-indeling is grotendeels dezelfde als bij $D01A; bij $D019 wordt echter ook nog bit 7 gebruikt. Bit 7 is altijd geset op het moment dat er een IRQ heeft plaatsgevonden.
In een eigen rasterinterrupt-routine zul je vaak een bewerking met adres $D019 vinden, deze is nodig om aan te geven welke IRQ je op dat moment afhandelt. Om een IRQ-vlag te resetten moet er een 1 worden geplaatst in het bit dat hoort bij de opgetreden IRQ. In het algemeen wordt voornamelijk bit 0 van $D01A en $D019 gebruikt. Dit om ervoor te zorgen dat je steeds op een bepaalde rasterregel terecht komt.

$D011 en $D012

Read/Write Raster value for Compare IRQ
Aan de hand van de adressen $D011 en $D012 kunnen we aangeven op welke rasterlijn er een IRQ moet plaatsvinden (De RASTERINTERRUPT). Hierdoor kunnen we de hele verdere opbouw van het scherm zelf beïnvloeden. We kunnen dan bijvoorbeeld voorkomen dat er flikkeringen optreden op het scherm. $D012 representeert de lowbyte van de rasterregel en geeft bit 7 van adres $D011 de 'most significant bit' van de rasterregel. Dit 'most significant bit' is nodig omdat het scherm 312 lijnen telt en dit getal past dus niet in één byte. Daarom wordt bij alle rasterlijnen boven de 256 bit 7 van $D011 geset. Door een waarde in deze adressen te poken geven we dus aan waar een RASTERINTERRUPT moet plaatsvinden. Na het inpoken is het niet mogelijk om de geplaatste waarden terug te lezen. Dit komt doordat deze twee adressen ook nog een andere functie hebben. Deze functie is het bijhouden van de huidige rasterregel. Dus als je de twee adressen uitleest krijg je de huidige rasterlijn die de computer aan het opbouwen is. We krijgen dus niet de waarden die we zelf in deze adressen geplaatst hebben.
Adres $D011 heeft naast bit 7 natuurlijk ook nog zeven andere bits. Deze hebben ook nog functies, maar die zijn in eerste instantie niet van belang bij het programmeren van rasterinterrupts.

$DC0E

CIA Control Register A
Normaal vinden IRQ's elke 1/60-ste seconde plaats. Dit wordt geregeld door de CIA-chip. Op adressen $DC04 en $DC05 wordt een teller bijgehouden (TIMER A). Wanneer deze teller 0 bereikt, wordt er een IRQ gegenereerd en de teller wordt een waarde gegeven die ervoor zorgt dat het 1/60-ste seconde duurt voordat hij weer nul bereikt. We kunnen deze timer besturen met adres $DC0E. We willen tijdens een rasterinterrupt liever niet onderbroken worden door een gewone IRQ, want dit zou vervelende gevolgen kunnen hebben. Om dit te voorkomen kunnen we met adres $DC0E aangeven dat de timer moet stoppen. Hiervoor gebruiken we bit 0 van adres $DC0E. Als dit bit 0 geset wordt zal de timer stoppen en er zullen geen gewone IRQ's meer plaatsvinden omdat de timer nooit meer de 0 zal bereiken. $DC0E heeft ook nog andere functies maar die zijn voor ons nu niet van belang.

We hebben de hierboven besproken adressen samengevat en in een tabel geplaatst. In deze tabel is kort aangegeven wat elk bit van het betreffende adres voor functie heeft. Er staan in de tabel twee adressen die nog niet besproken zijn. Deze adressen zullen in een latere aflevering nog besproken worden.
Het voorgaande was aardig taaie theorie en we hebben nu wel de adressen besproken, maar hoe we ze nu precies

moeten gebruiken zal nog steeds niet echt duidelijk zijn. We zullen dit nu proberen te verduidelijken aan de hand van listing 1. Deze listing is geschreven in 'turbo-assembler' en voor iemand die een klein beetje ervaring heeft met assemblers en geen 'turbo-assembler' heeft hoeft het geen probleem te zijn deze listing om te zetten.

Simpel

Deze listing is een simpel voorbeeld van een rasterinterrupt. Laten we het programma eens even nagaan.



We beginnen met het initialiseren van de interrupt. Daarvoor zetten we de IRQ even uit. Vervolgens veranderen we de IRQ-pointer naar een eigen routine. Door de timer A via adres $DC0E uit te zetten zullen er tevens geen normale IRQ's meer plaatsvinden. In $D011 en $D012 wordt dan een waarde geplaatst die aangeeft op welke rasterlijn de raster-IRQ zal gaan plaatsvinden. Daarna wordt door middel van de CLI het interrupt-verkeer weer toegestaan.

In de nieuwe IRQ-routine wordt er begonnen met het resetten van het vlag-register, zodat na de afronding van deze routine er weer een nieuwe IRQ kan plaatsvinden.

Het volgende stukje programma zorgt ervoor dat vanaf rasterlijn $33 de eerst komende 32 rasterlijnen een door ons zelf gekozen kleur krijgen. Dit is vrij makkelijk. We poken een nieuwe kleur in $D020/$D021 aan het begin van de rasterregel en wachten een tijdje totdat de computer de volgende rasterregel bereikt heeft. Vervolgens plaatsen we aan het begin van de volgende rasterregel weer nieuwe kleuren. Dit doen we een aantal keren achter elkaar, waardoor een zogenaamde 'RASTER-BALK' onstaat. Door de waarden van de tabel 'KLEUREN' te veranderen kan je andere kleurenbalken doen ontstaan.

Cycli

Het wachten heeft betrekking tot het aantal cycli dat het duurt om één rasterregel op te bouwen. Door de wachtlus duurt het één keer doorlopen van de routine iets minder dan 64 cycli want de computer gebruikt zelf ook nog wat tijd waardoor, alles opgeteld, het één keer doorlopen van de routine ongeveer 64 cycli kost. Dit is dus ongeveer één rasterregel (zie vorige aflevering). Deze routine is niet 100% nauwkeurig, dit is te zien als je de vertical-hold van de monitor iets verandert. Er zijn dan links of rechts zogenaamde rasterbugs te zien (Kleine flikkeringen in de lijnen). Dit wordt veroorzaakt door het feit dat de routine niet altijd 64 cycli doorloopt. De schermkleur wordt dan te

Een voorbeeld van een typische rasterinterrupt.

```
     *= $1000
;-------------------------------------
     SEI         ;VOORKOMT IRQ'S.

     LDA #
   Q  ;HIER WORDEN DE IRQ
     STA $0314   ;VECTOREN VERANDERD.
     LDA #IRQ
     STA $0315

     LDA #$01    ;LAAT COMPUTER OP RASTER
     STA $D01A   ;IRQ REAGEREN.
     LDA #$00    ;SCHAKEL TIMER A UIT,DUS
     STA $DC0E   ;GEEN TIMER-IRQ MEER.

     LDA #$33    ;BIJ RASTERLIJN $033
     STA $D012   ;MOET EEN RASTER-IRQ
     LDA $D011   ;PLAATSVINDEN.
     AND #$7F
     STA $D011

     CLI         ;LAAT WEER IRQ'S TOE.
     RTS

IRQ
     LDA $D019
     ORA #$01    ;RESET RASTER-IRQ-VLAG
     STA $D019

     LDX #$00
A1   LDY WACHTLUS,X
     DEY
     BNE *-1
     LDA KLEUREN,X
     STA $D020    ;BORDERKLEUR
     STA $D021    ;SCHERMKLEUR
     INX
     CPX #$20
     BNE A1

     JMP $EA31    ;VERVOLG NORMALE IRQ

;-------------------------------------
WACHTLUS
      .BYTE 6,8,8,8,8,8,8,1
      .BYTE 8,8,8,8,8,8,8,1
      .BYTE 8,8,8,8,8,8,8,1
      .BYTE 8,8,8,8,8,8,8,1
;-------------------------------------
KLEUREN
      .BYTE 2,0,2,10,0,2,10,7
      .BYTE 0,2,10,7,1,0,2,10
      .BYTE 7,1,1,3,1,3,3,14
      .BYTE 3,14,14,6,14,6,6,0
;-------------------------------------
```

Listing 1



vroeg veranderd waardoor de flikkeringen onstaan. We zullen in latere afleveringen terug komen op dit probleem. Via de JMP $EA31 wordt ervoor gezorgd dat de computer zijn gewone IRQ routines afrondt. En dat is alles. De in deze routine gebruikte initialisatie (de eerste 15 regels) is een soort standaardvorm voor raster-interrupts. We zullen deze initialisatie dan ook nog wel vaker tegenkomen in toekomstige voorbeeld listings. Zo ook in listing 2.

Rasterbalk

Dit is een toepassing van de eerder gegeven listing. We hebben hier een routine geschreven die een 'Rasterbalk' over het scherm op en neer beweegt. Je kunt natuurlijk deze routine zelf zo wijzigen dat de balk een baan beschrijft die je zelf hebt gekozen. Je zou de balk bijvoorbeeld volgens een sinus kunnen laten bewegen. Om dit te bewerkstelligen moet je de routine 'BEWEEG' herschrijven, je zou hier bijvoorbeeld een tabel kunnen uitlezen waarin alle posities voor de balk zijn opgeslagen. Deze positie moet dan in de variabele 'PUNT1' geplaatst worden en waarneer de routine 'ZETBALK' dan wordt aangeroepen, zal de balk op de nieuwe positie worden geplaatst. Je kunt met de gegeven tabel 'WACHTLUS' momenteel een maximum van $28 op en neer bewegen. Een andere toepassing zou het draaien van zo'n 'rasterbalk' kunnen zijn, en zo zijn er nog tal van dingen te verzinnen.

Ansgar Smith & Wytze B. Westra

Een toepassing van de eerder gegeven rasterinterrupt.

```
     *= $1000

     SEI          ;VOORKOMT IRQ'S.
     LDA #
   Q  ;HIER WORDEN DE IRQ
     STA $0314  ;VECTOREN VERANDERD.
     LDA #IRQ
     STA $0315
     LDA #$01   ;LAAT COMPUTER OP RASTER
     STA $D01A  ;IRQ REAGEREN.
     LDA #$00   ;SCHAKEL TIMER A UIT,DUS
     STA $DC0E  ;GEEN TIMER-IRQ MEER.
     LDA #$33   ;BIJ RASTERLIJN $033
     STA $D012  ;MOET EEN RASTER-IRQ
     LDA #$1B   ;PLAATSVINDEN.
     STA $D011
     LDA #$00
     STA PUNT1
     STA PUNT2
     JSR LEEG
     CLI          ;LAAT WEER IRQ'S TOE.
     RTS
IRQ
     LDA #$01   ;RESET RASTER-IRQ-VLAG
     STA $D019
     NOP
     LDX #$00
A1   LDY WACHTLUS,X
     DEY
     BNE *-1
     LDA KLEUREN,X
     STA $D020   ;BORDERKLEUR
     STA $D021   ;SCHERMKLEUR
     INX
     CPX #$30
     BNE A1

     JSR LEEG    ;MAAK KLEUREN LEEG
     JSR ZETBALK;ZET BALK OP NIEUWE PLEK
     JSR BEWEEG ;+/- POSITIE BALK
     JMP $EA31   ;VERVOLG NORMALE IRQ

LEEG
     LDY #$2F
     LDA #$00
L1   STA KLEUREN,Y
     DEY
     BPL L1
     RTS
ZETBALK
     LDX PUNT1
     LDY #$06
Z1   LDA BALK,Y
     STA KLEUREN,X
     INX
     DEY
     BPL Z1
     RTS
BEWEEG
     LDA PUNT2
     BNE B1
     LDX PUNT1
     CPX #$28
     BEQ B2
B4   INC PUNT1
     RTS
B1   LDX PUNT1
     BNE B3
     DEC PUNT2
     JMP B4
B2   INC PUNT2
B3   DEC PUNT1
     RTS
;--------------------------------------
     *= $1200
WACHTLUS
     .BYTE 6,8,8,8,8,8,8,1
     .BYTE 8,8,8,8,8,8,8,1
     .BYTE 8,8,8,8,8,8,8,1
     .BYTE 8,8,8,8,8,8,8,1
     .BYTE 8,8,8,8,8,8,8,1
     .BYTE 8,8,8,8,8,8,8,1
BALK
     .BYTE 6,4,3,1,7,10,2
PUNT1 .BYTE 0
PUNT2 .BYTE 0
KLEUREN
     .BYTE 0
```

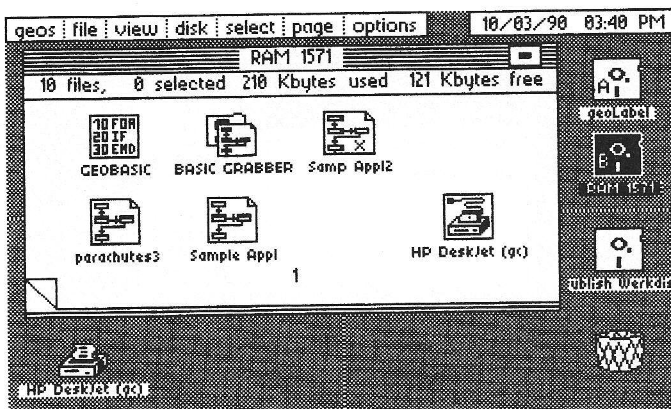
Listing 2



GEOBASIC

Eigen programma's onder GEOS

Van de week kreeg ik een brief in de bus van de Douane. Er lag een pakketje uit Amerika voor mij op het hoofdkantoor van de PTT in Amsterdam. Dat zou dus geoBasic moeten zijn want verder had ik in Amerika geen bestellingen meer lopen.



Douane inklaring

Ik naar Amsterdam en nadat het pakje door hen was opengemaakt en gekontroleerd was of er geen drugs tussen de bladzijden van het boek was blijven zitten kon ik invoerrechten en de BTW betalen. Opgetogen vertrok ik met geoBasic. Ik had er lang genoeg op moeten wachten. Thuis gekomen kon ik natuurlijk niet wachten. Het bureau heb ik met de bekende beweging schoongemaakt en toen had ik ruimte om met geoBasic aan de gang te gaan. Mijn C128 snorde tevreden terwijl GEOS opstartte.

Zodra de deskTop op het scherm verschijnt, gauw de geoBasic disk in de gleuf en de inhoud met **C= K** snel gekopieerd naar mijn RAM. Het geoBasic icon heeft een bekend gezicht. Basic regels. Een beetje afgezaagd misschien, maar wel doeltreffend.

System error

Ik dubbelklikte het geoBasic icon en kreeg natuurlijk de melding dat ik terug moest schakelen naar 40-koloms mode. Dus ik klik op YES en zet de schakelaar op 40, om getuige te zijn van een dialoog box met de mededeling dat ik een **system error near $4755** heb. Bedankt, ik heb eigenlijk nog steeds niet begrepen wat deze adressen inhouden. Inwendig vervloekte ik de Douane ambtenaar die misschien te enthousiast was geweest met het zoeken naar drugs, en daarmee de disk misschien had beschadigd. Of met een magneet had gekeken of er wel genoeg ijzerdeeltjes op de disk aanwezig waren geweest. Terug op af dus.

Er zat gelukkig een boek bij de disk dus laten wij daar dan maar eerst in kijken. Zo'n boek heb je als GEOS gebruiker toch eigenlijk niet nodig. Maar ja, het zit er niet voor niets bij. De dikte van het boekje viel mij eigenlijk een beetje tegen. Vergelijkend met de pil die bij geoProgrammer wordt bijgeleverd. Op de eerste bladzij kom ik naast de uitleg dat geoBasic tot stand is gekomen door de samenwerking van Berkeley en de uitgeverij van het Commodore maandblad RUN Magazine, waarvan RUN verantwoordelijk is voor de handleiding, de melding tegen dat "**geoBasic runs in C64 mode(40 colomns) on C64/128.**".

Met andere woorden, er is weer een applikatie die niet gebruik maakt van het 80 koloms scherm van onze goede oude C128.

Op zoek naar de disk GEOS 2.0 voor de C64. Ik moest er nog ergens een hebben. Terwijl ik zoekende ben gaat mijn deurbel. De Postbode.....

Markt & Technik

De goede man brengt al een tijdje bij mij de post en vertelt mij nu dat het hem opviel dat ik zoveel diskettes ontving en nu had hij de 64er bij zich. Ook daar ben ik op geabonneerd, maar dit even terzijde. Aan de 64er had hij gezien dat ik waarschijnlijk met een Commodore werkte en hij had er ook een.

Kompleet met datasette en genotsknots(joystick). Ik vroeg of hij even tijd had voor een kopje koffie en een demonstratie echt (GEOS) computeren. een ander keertje zei hij en liet mij achter met de post. Terug achter mijn C128 met de post keek nog eens naar de system error melding en dacht: Eerst maar koffie en de 64er.

Uit de envelop komt het septembernummer en blijkt een verslag over geoBasic door Paul Zettler te hebben. Dus interessant genoeg om even te lezen, voordat ik weer verder zou gaan met mijn eigen worsteling.

Het verslag ging over Duitse versie van geoBasic. Zoals u wellicht al wist is Mark & Technik de uitgeverij die de GEOS produkten van Berkeley onder licentie in het Duits uitbrengt.

We kennen inmiddels allemaal het bekende deskTop probleem.

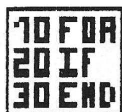
Duitse versies van de verschillende applikaties vragen allemaal om de Duitse deskTop en accepteren niet de Amerikaanse deskTop. Om een lang verhaal kort te maken, geoBasic is niet vertaald naar het Duits. Het handboek wat erbij geleverd wordt is uiteraard wel in het Duits. Wat betreft het verslag, tussen de regels door lees je dat als je geen BASIC beheerst het toch heel moei-



lijk is om alle begrippen die in geoBasic gebruikt worden voor de volle 100 procent te begrijpen. Het programma blijkt toch niet zo eenvoudig. Zeker er zijn krachtige editors die je een bepaald gemak kunnen geven. Je kunt het een beetje vergelijken met het Simon's Basic van een aantal jaren geleden. Er wordt duidelijk gesteld dat als u nog nooit een basic programma(tje) heeft gemaakt het werken met geoBasic wordt afgeraden. Nu zijn er genoeg boeken in de winkel verkrijgbaar waarmee u uw kennis van BASIC kunt opfrissen, of in elk buurthuis is er tegenwoordig een mevrouw of meneer die een BASIC cursus geeft. Doe zo'n cursus. Voor de prijs hoeft u het zeker niet te laten. De kennis die u opdoet is dan nog zo fris dat u waarschijnlijk met geoBasic programma's kunt maken waarmee u uw BASIC leraar mee om de oren kan slaan.

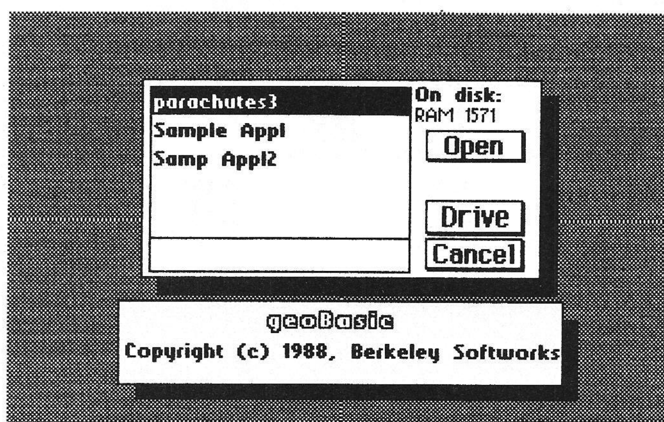
Duitse handleiding

Wat mij tevens opviel was dat ik het idee had dat de Duitse handleiding zeer uitvoerig moest zijn want er werd over voorbeelden, tekeningen en schermafdrukken gesproken. Het boek dat bij de Amerikaanse versie van geoBasic zit vertoont geen enkele tekening of schermafdruk. Dus leek mij het een goede zaak als ik ook de Duitse versie zou aanschaffen en de Amerikaanse versie even te laten voor wat hij was.
De eerste gelegenheid die zich voordeed heb ik gebruikt om in Essen even een boekwinkel binnen te stappen en de Duitse versie van geoBasic aan te schaffen. Voor Dm 89,-- werd ik eigenaar van geoBasic. Aanzienlijk minder duur dan dat ik uiteindelijk voor de Amerikaanse versie moest betalen. Ik had het ook sneller in mijn bezit. De handleiding ziet er inderdaad goed verzorgd uit en er zaten zelfs twee diskettes in.



geoBasic

tek.2Terug naar geoBasic. Ik heb inmiddels mijn 64 GEOS disk gevonden en in de 64 mode GEOS geladen en de Duitse geoBasic naar RAM gekopieerd.



Nogmaals dubbelklikken op het geoBasic icon brengt mij naar het openingsscherm met de CREATE/OPEN/ QUIT opties. GeoBasic zelf was zoals ik reeds vermoedde niet in het Duits vertaald. Er stond QUIT op het scherm in plaats van het Duitse ABBRUCH. Het maakt dus geen verschil of u nu de Amerikaanse versie of de Duitse versie aanschaft, werken doet het altijd. Om voor de keuze CREATE te kiezen vond ik het nog een beetje vroeg. Dus dan maar OPEN.
Parachutes leek mij wel wat. He, dat lijkt de Commodore karakterset wel. Die dikke hoekige letters die had ik lang niet gezien. En READY. En het knipperende blokje dat de cursor moet voorstellen. Even list tikken. Dat werkte. er staat een listing op het scherm. Maar dit is geen basic, of toch......

```
1010 @ChDr2:
1020   sprt
     5,-4,g:IF g220
     THEN delproc
     @ChDr2: GOTO SP2
1030 RETURN
```

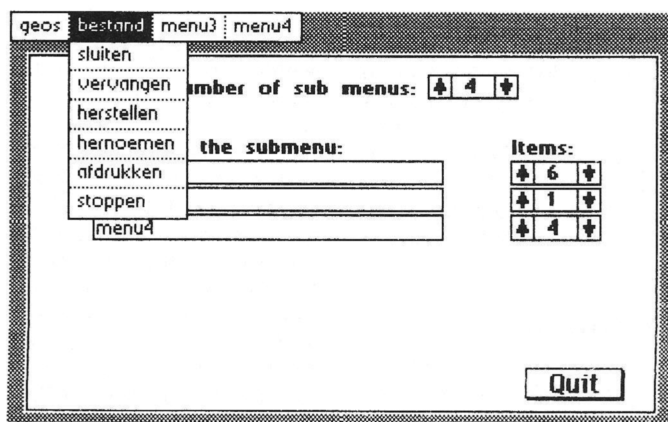
Nu zie ik ook een REM regel staan die mij verteld dat bovenstaande routine checkt of iets de grond raakt. Als mijn kop koffie op de grond valt dan hoor ik dat en heb daar dus bovenstaande routine niet voor nodig. Dus het boek erbij......

In regel 1010 staat een LABEL. Dat noemen ze zo. Een label is een aantal instrukties die je onder een naam kunt samenvoegen.
De instruktie is sprt 5,-4,g(sprt staat voor een bewegende sprite) en deze instruktie geven wij de labelnaam ChDr2.
Om geoBasic te laten weten dat dit een label is zetten wij er een @(apestaartje) voor. @ChDr2 dus. Simpel he.
De IF-THEN instruktie weten wij nog wel, maar dan gaan we weer
met delproc @ChDr2. Delproc is de stop instruktie van process.
Dus eerder in de listing zal de instruktie process @ChDr2 staan. De GOTO en RETURN zijn weer normale BASIC instrukties.
Nu gaat het mij te ver om hier en nu een komplete cursus geoBasic te geven, daar het te meer gaat om een globaal overzicht van geoBasic.

Buiten het feit dat geoBasic gebruik kan maken van Labels, zijn er nog een aantal krachtige instrukties die wij bij het gewone Commodore Basic missen. Wij krijgen nu de kommando's die wel in C128 Basic 7.0 bekend zijn als REPEAT, UNTIL en WHILE...ander zijn krachtige Monitor kommando's zoals ONERR, DEBUG en RDBYTE en voor de besturing van de SID chip de kommando's SOUND en VOICE. Zoals ik al zei; het heeft een beetje weg van Simon's Basic.

Editors

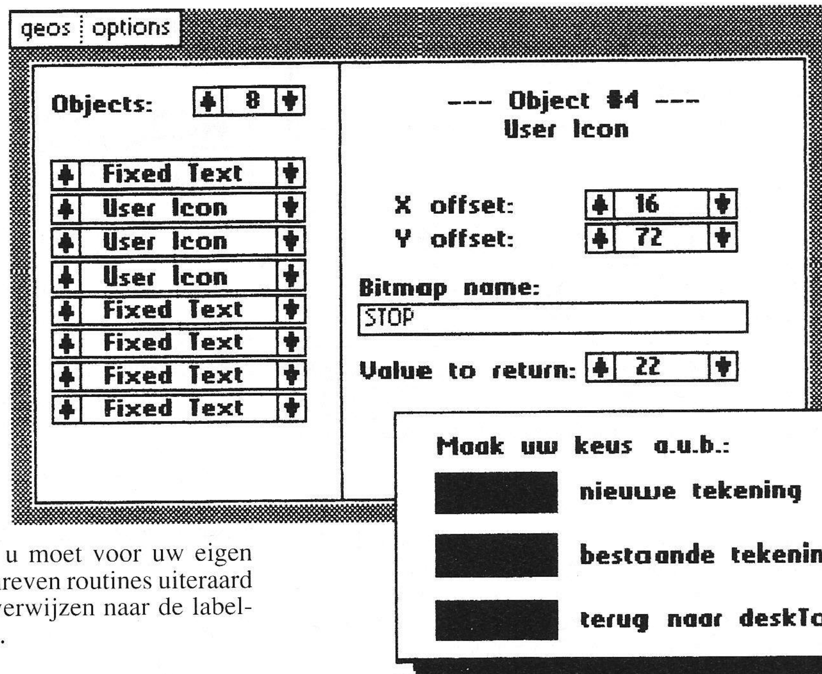
GeoBasic heeft 5 Editors aan boord. De Menu-, dialoog-





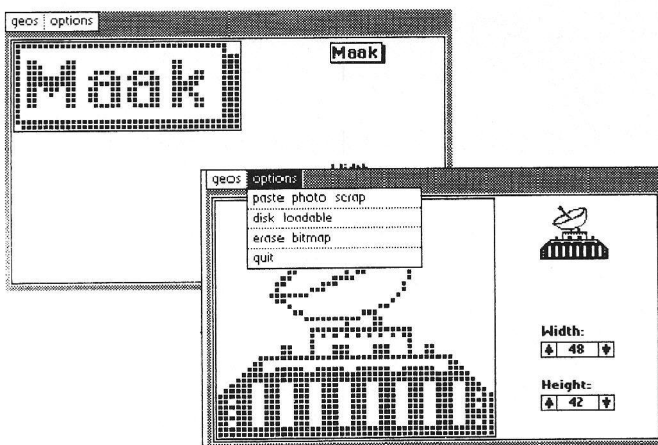
box-, Icon-, Bitmap- en Sprite Editor. De menu- en dialoogbox Editor zijn wel de krachtigste van het stel. Hiermee kunt u uw programma dat specifieke GEOS uiterlijk mee geven. In plaats van het bekende menu te maken zoals u dat in de Basic cursus leert, met keuze opties, krijgt u het GEOS menu ervoor terug.

Met de Menu Editor kunt u zelf het kommando menu samenstellen. U krijgt de opdracht om aan te geven uit hoeveel opties deze moet bestaan. En nu het krachtigste: en in welke taal......
Want op het moment dat u het aantal opgeeft verschijnt het kommando menu op het scherm met geos menu2 menu3 etc.
De optie geos is standaard aanwezig en kan niet veranderd worden. Maar wie zou dat dan ook willen.
Geef nu aan hoeveel submenu's ieder optie moet bevatten. Daarna kunt u deze opties invullen. Voert u een optie in dan wordt u ook gevraagd om de **GOSUB** labelnaam voor het programma op te geven bijv: u geeft in een submenu de keuze **hernoem**.
De labelnaam die daaraan verbonden zit is **@DoRename**.
Deze labelnaam zult u dan ook moeten opgeven. Dit is dan een bestaande labelnaam maar u moet voor uw eigen geschreven routines uiteraard ook verwijzen naar de labelnaam.



Dialoogbox Editor

De dialoogbox Editor werkt in feite hetzelfde als de Menu- Editor. Alleen zijn hier een aantal teksten standaard al aanwezig. Het **OK**, **CANCEL** en **QUIT** blok zijn een voorbeeld hiervan. De dialoogbox Editor is ook de Editor waarin uw de invoer vanaf het toetsenbord kunt regelen. In de dialoogbox die u aanmaakt kunt u zowel tekst als graphics plaatsen. Met zelfgemaakte iconen kunt u een bepaalde keuze kracht bijzetten.



Bitmap- en Sprite Editor

Met de Graphic Editor kunt u zelfgemaakte tekeningen later in het programma invoegen. Met deze editor is het mogelijk om in twee kleuren te werken. Meer gaat niet. Wilt u meer, dan zult u in geoPaint moeten gaan werken en daar dan een photoscrap van maken.
Deze photo-scrap kan dan wel verder worden verwerkt. De maximale grootte van een graphic is 48 pixels breed en 42 pixels hoog. Ook heeft u de keuze om deze graphic variabel te maken voor meervoudig gebruik of als vaste data op disk. Als u kiest voor het laatste dan kunt u deze data niet meer gebruiken.

Met de Sprite Editor heeft u maximaal 6 (3-8) sprites tot uw beschikking. De eerste twee sprites heeft GeoBasic nodig voor het pijltje en de tekstcursor. De maximale grootte van een sprite is 12 bij 21 pixels. Voor grotere sprites dient u gebruikt te maken van de **link sprites** optie.

Heeft u de Sprite Editor gestart dat heeft u de keus om de kleur, start positie, snelheid en spritenummer te kiezen.
Via het **EDIT** icon komt u in de eigenlijke Editor.
Hier kunt u dan uw sprite aanmaken. In het kommando menu staat de optie Options. In dit submenu heeft u weer de keus om sprites te linken en met de **paste frame** optie is er zelfs **animatie** mogelijk.

Icon Editor

Deze icon Editor moet u niet verwarren met de Icon Editors die er al zijn. Deze Icon Editor is om icons te maken zoals de tekenopties in GeoPaint. Klikt u op zo'n icon dan heeft dat een reaktie tot gevolg. Ook in deze Icon Editor heeft u de keus om een nieuw of bestaand icon te wijzigen. Maakt u een nieuw icon aan dan krijgt u de opdracht om het icon nummer aan te geven (u heeft er maximaal 31 tot uw beschikking) de X- en Y-positie de Bitmap naam en de **Place to go**



geos | options | new...

Sprite number: 4

TRAIL to X: 180 to Y: 230

TRAIL speed: 48 At end, gosub:
Stop Motion at trail end ☒ @Explode

ANIMATION Frames: 2 Continuous ANIMATION ☒
ANIMATION Rate: 8 Edit

(GOTO opdracht in het programma) instruktie. U kunt hier het regelnummer van het programma neerzetten. Beter is om alle programma routines een labelnaam te geven. U hoeft dan alleen de labelnaam neer te zetten. Verandert u later in het programma een regel dan loopt u geen risico dat het programma crasht.

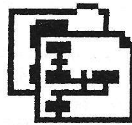
De Debugger

Over crashen van het programma gesproken. Niemand is perfect en ik geloof ook niet dat iemand wel eens een programma heeft gemaakt dat in een keer liep. Daarom is in het Options menu van GeoBasic een **Debugger** aanwezig. In het dialoogvenster van deze Debugger kunt u de benodigde parameters ingeven.

Met **OK** start u dan de Debugger. Komt deze dan een onrechtmatigheid tegen zal deze het Basic regelnummer en de waarden naast de door u ingestelde parameters weergeven.

Tevens heeft het Debugger dialoogvenster twee submenu's.
In het **mode** submenu vindt u de opties **Run, step** en **trace**. Met **Run** start u het programma en komt de debugger alleen in aktie als er iets fout gaat. Alle door u opgegeven waarden worden dan getoond. Met **step** kunt het programma stapsgewijs doorwerken. Na iedere regel verschijnt het dialoogvenster met de waarden. Met **OK** kunt u de volgende stap doen. En **trace** werkt als **run**, alleen wordt er gestopt bij een door u aangebracht breakpoint.

Debug | mode | show
numSprs 5
a 1.7570393
i 41
@man1 440
OK Line 340 Cancel



Basic Grabber

Op de diskette staat nog een programma. **Basic Grabber** is de naam van dat programma. Het doet ook precies wat het zegt. Het "grabt Basic".
Basic Grabber lijkt in vele opzichten op de Text- Grabber die u waarschijnlijk allemaal al eens gebruikt heeft. Alleen kunt u nu bestaande Basic programma's inlezen. Basic Grabber zal het bestaande programma omzetten naar een formaat dat door geoBasic kan worden begrepen. Tijdens deze konversie zal Basic Grabber de instrukties die hij niet begrijpt of niet kan omzetten eerst op het scherm laten zien. Door middel van **OK** kunt u doorgaan en zal Basic Grabber voor deze niet begrepen instruktie een REM plaatsen.
Peek's en Poke's worden zonder meer niet vertaald. Dit soort instruktie moeten dan ook worden vermeden.

source: DELTADRES1 output: Deltadres
The token INPUT#
OK Cancel
Basic Grabber
Copyright (c) 1988 Berkeley Softworks

Tot slot

Ik denk zelf dat door de komst van geoBasic de drempel tot het zelf programmeren weer met enkele treden is verlaagd.
Nu kunnen wij zelf onze ideeen gaan uitwerken en met behulp van geoBasic tot een krachtig geheel vormen.
Toch zijn er enkele kanttekeningen te plaatsen. Door de forse ruimte (77 Kb) die geoBasic al nodig heeft blijft er weinig ruimte over om snel te kunnen werken. Programma's die de 10 Kb (40 blokken) inclusief variabelen niet overschrijden kunnen snel genoemd worden.
Het is daarom belangrijk om gestruktureerd te programmeren.
Maak veelvuldig gebruik van labels. Heeft u eenmaal een bepaalde routine een keer gemaakt geef deze dan een labelnaam. Zet deze routine op een disk waar u ook andere routines op heeft staan die al een labelnaam hebben. U kunt dan zo een hele bibliotheek opbouwen. De praktijk leert ons dat er een heleboel te doen is met slechts enkele komponenten (routines).
Voorwaarde is dat u Basic redelijk goed beheerst.
Hopelijk kunnen wij binnenkort programma's zien die nu nog niet onder GEOS draaien maar door uw toedoen dat wel doen.
Laat ons dit dan weten.
Veel succes en ga een uitdaging aan met geoBasic.

Pim Broekhuijzen



GEOS-INFO

Voor deze aflevering was er weer een aardige hoeveelheid post. De r is nu in de maand en alweer een nieuw computerseizoen staat dus te popelen om te beginnen. Hoewel teruggekeerd in Nederland, heeft Bert Venema het nogal druk, vandaar dat ik de GEOS INFO-rubriek weer onder mijn hoede heb genomen.

Dhr. Bruin uit Haarlem meldt ons dat hij onlangs een tweedehands C-128 gekocht heeft. Bij deze aankoop heeft hij ook een GEOS 2.0-pakket overgenomen met originele schijven. Dhr. Bruin bezat al een 1541-drive. Bij het booten begint de drive echter na de melding "booting geos" te ratelen, en stopt er dan mee. Op de 1541 van de eerdere eigenaar werken de GEOS-schijven echter wel. Zijn vraag is nu: waar ligt het aan en wat kan ik er aan doen?

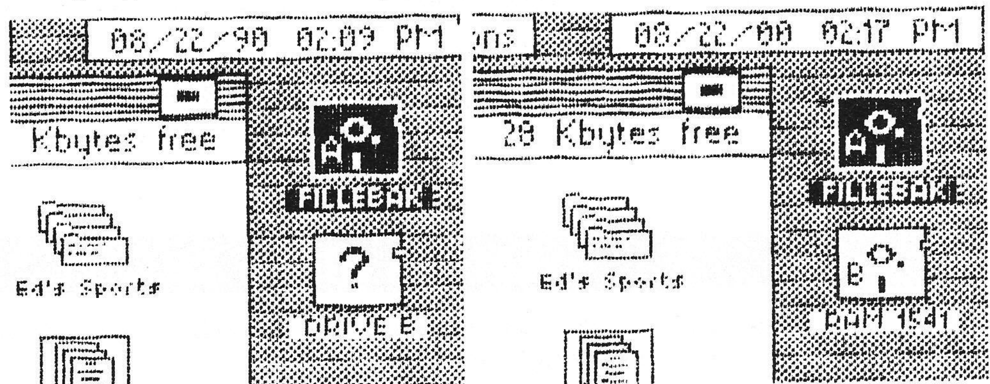
Kopstand

Het probleem van dhr. Bruin lijkt mij een typisch geval van het verkeerd (of liever gezegd: verschillend) staan van de read/write-kop in de 1541 drive van dhr. Bruin. Er zijn enkele programmaatjes in omloop die door de kop van track 1 naar 35 heen en weer te smijten proberen om de kopstand te veranderen. Het effect van dit soort programmaatjes is een beetje dubieus, maar je kunt het altijd proberen. Een meer solide oplossing is om bij een vakhandelaar (of bij uw lokale C-64 whiz-kid) de kop opnieuw af te laten stellen.

A.C. Wille komt met een ouderwets echt printerprobleem. Hij heeft een Commodore MCS 801 (een kleurenprinter) en er is geen driver die daarmee wil werken. Zonder het nu over kastjes naar de muur te willen hebben verwijs ik dit probleem graag door naar de Stichting GEOS gebruikers in Almere. Zij hebben daar ervaring met een heleboel printers en met een nog grotere boel vragen daarover. Misschien bezitten ze al een MCS 801-driver (ik heb er geen), en als ze hem niet hebben beschikt de Stichting GEOS gebruikers, naar ik begrepen heb, over software om gemakkelijk een driver in elkaar te draaien. Of deze dan in kleur zal werken, lijkt mij overigens wel erg twijfelachtig.

Geofile

P. Havenaar vertelt het volgende verhaal: als ik het programma GEOFILE op de gebruikelijke wijze opstart, na de normale opstartprocedure van GEOS BOOT waar ook de datum en de tijd gezet worden, en er is een minuut of 20 gewerkt en ik ga terug naar de DESKTOP, dan is de aanduiding van het jaartal weg en staat op 00.



De enige verklaring hiervoor lijkt mij een bug(je) in GEOFILE. Kennelijk wist GEOFILE bij het wegschrijven van zijn variabelen het adres $8516 in de kernal. Op dit adres en tot 5 bytes verder staan namelijk de datum, jaar en tijd opgeslagen.

Petro Kooy uit Bleiswijk vraagt of er een programma bestaat dat de listings uit Run Magazine of Compute Gazette weer terug kan converteren naar een Geowrite-file zoals van toepassing in de GeoProgrammer. De listings in Run Magazine en Compute Gazette (beide overigens erg interessant op GEOS-gebied) hebben de vorm van dataregels. Een programmaatje in de trant van mijn eigen 'GEOS PROGRAM-MAKER' (uit een heel aantal afleveringen geleden) schrijft die dataregels dan weg met een fileheader, zodat er een GEOS-applicatie ontstaat.

Disassembler

Het antwoord op Petro's vraag is echter: nee, zo'n programma bestaat niet. Het is immers nooit mogelijk om de label-namen uit een assembly listing terug te vinden. Het enige wat je achteraf kan doen is disassembleren, en dat is niet altijd even begrijpelijk. Voor zover mij bekend bestaat er geen GEOS-disassembler. De GeoDebugger is weliswaar een disassembler, maar daar heb je altijd een '.dbg'-file voor nodig naast de te debuggen file. En dat is een zware beperking. Maar als iemand weet van het bestaan van een GEOS (PD)-disassembler, laat het dan eens weten aan deze rubriek!

Tevens doet Petro Kooy verslag van een merkwaardig probleem: het programma uit deel 11 van de machinetaalcursus, de GeoPaint Scanner, werkt bij hem niet. Als hij een menu-item aanklikt, dan licht alleen de meest linkse optie op, dat is 'write'. Dit gebeurt dus ook als een andere optie wordt aangeklikt! Waar het aan ligt is onduidelijk. De conclusie dat zijn menutabel in het interne geheugen tijdens runtime kennelijk gemold wordt, dringt zich op. Als Petro mijn menutabel vervangt door een door hem geschrevene, die wat meer naar rechts staat,



dan werkt alles echter wel! Maar mijn menutabel staat goed in het blad en werkt op mijn C-64. Aangezien ik er niemand anders over gehoord heb neem ik maar aan dat ik het inderdaad bij het rechte eind heb, en dat de listing goed is. Petro heeft of toch ergens een tikfout gemaakt, of er zit een onwaarschijnlijke bug in z'n Geo Assembler. Als er echter meer mensen zijn met dit probleem, geeft u dit dan a.u.b. door, want dan ligt het toch waarschijnlijk aan mijn listing.

1571-drive

Peter Buijsman uit Beverwijk werkt al 3 jaar met GEOS en heeft naar eigen zeggen een hechte band met het pakket. Hij is in het bezit van de (relatief oude) versie 1.3. Hij heeft onlangs zijn C-64 omgeruild voor een C-128D met twee 1571-drives. Op beeldende wijze beschrijft hij mij in zijn brief, hoe hij geprobeerd heeft om GEOS aan de praat te krijgen met de 1571-drives in de 1571-mode, en dus niet in de gesimuleerde 1541-mode. Het voordeel van het werken met de 1571 in de eigen mode, is dat er dan met dubbelzijdige disks gewerkt kan worden zonder het omdraaien van de diskette, aangezien een 1571 twee read/write-koppen heeft.

Het laten werken van de 1571 in de dubbelzijdige mode is echter vrijwel onmogelijk met GEOS 1.3. De GEOS Kernal maakt namelijk van de DiskTurbo snellaad/save-routines gebruik, en deze zijn op maat gemaakt voor de 1541-drive. En het herschrijven van de kernal is natuurlijk geen echt reële mogelijkheid.

Foutje

In de vorige aflevering van de machinetaalcursus staan in de LIST 3 enkele foutjes. Deze zijn veroorzaakt doordat het tekst-opmaak-programma (Ventura) van de uitgeverij groter-dan en kleiner-dan tekens in de listing aanzag voor control-tekens. De vijfde regel moet zijn:

```
.if (value>=0)&&(value<32768)
```

Tevens is de laatste regel van LIST 3 weggevallen, dit was ".endm" (zonder de aanhalingstekens natuurlijk). Tot slot van deze GEOS INFO-rubriek zal ik u nog een tip van de hand doen: haal nooit een seriële kabel uit een drive als hij nog aanstaat. Vroeger wist ik dit ook wel, maar ik geloofde daar nooit iets van en mijn hele leven plugde ik dus al kabels in en uit drives als ze aanstonden. Tot vorige week zondag. Nu zijn allebei mijn drives kapot, en daardoor is het programma voor de machinetaalcursus deel 13 (ik had het kunnen weten) niet af, en staat die dus niet in dit blad. Dat programma (het is een 3D-programma, dat kan ik al verklappen) komt dus in de volgende, misschien wel laatste aflevering van GEOS-machinetaal. Maar wie weet moet ik deel 13 maar gewoon overslaan, want er kan nog heel wat kapot aan mijn computer.

Peter Boncz

AMIGA BUSWARE

Compleet assortiment Amiga PDS software voor ƒ 11,- per schijf. Vraag nu een gratis catalogus aan of bestel voor ƒ 11,- de speciale introductiediskette, namelijk:

de Amiga Busware Introschijf

Dit is een schijf uit het Busware assortiment, die we samen met een aantal Amiga specialisten hebben samengesteld. Daarop staat de volgende selektie:

Grafisch demopakket met workbench schermgrapjes, waaronder Wave Bench, Melt en Dropshadow
Gauge om te bepalen hoevel geheugen er vrij is
Record Player (om zelf demonstraties te maken, alle muis en toetsbewegingen kunnen herhaald worden)
Helios Mouse (maakt venster, waar de muis is, actief)
Pins (grafisch programma)
Asteroids (ruimtespel)
DOS kwik Om meer op schijf te krijgen
Drunken Mouse (de muis gaat rare bewegingen maken op het scherm)
Backgammon spelprogramma
X-Icon Om programma's, die geen ICON hebben, toch te kunnen starten met Icon
Conman (CLI Editor)

bel: 020-273198 /02152-62343



TIPS & TRUCS 64

Deze keer besteden Michel de Boer en Hylke Sprangers aandacht aan de machinetaal instructie set van de Commodore 64. Verder geven zij een idee waarmee teksten in een zelf-gedefinieerd lettertype op de printer kunnen worden afgedrukt.

De meeste mensen die een Commodore 64 bezitten kunnen wel in machinetaal programmeren. Het hiervolgende gedeelte van Tips en Trucs is bedoeld voor deze mensen. We gaan namelijk de gehele instructieset van de 6510-processor per instructie langslopen. Dat wil zegen dat we voor elke instructie kort aangeven wat deze doet, welke adresseringsmodi de instructie kent met daarbij de opcode, welke flaggen worden geset en hoeveel cycles de instructie vraagt. Deze Tips en Trucs is echter niet bedoeld als een bliksemcursus machinetaal. Het is meer bedoeld als een volledige naslagtabel. Hieronder volgt nog een verklaring van de gebruikte afkortingen.

N: Negative-flag
Z: Zero-flag
C: Complement-flag
I: Interrupt-flag
D: Decimal-mode-flag
V: Overflow-flag
imm: immediate-addressing
abs: absolute-addressing
zero: zeropage-addressing
accu: accumulator-addressing
impl: implied-addressing
(ind,x): (indirect,x)-addressing
(ind),y: (indirect),y-addressing
zero,x: zeropage,x-addressing
zero,y: zeropage,y-addressing
abs,x: absolute,x-addressing
abs,y: absolute,y-addressing
rel: relative-addressing
ind: indirect-addressing
OP: Opcode
NB: Aantal bytes
NC: Aantal cycles

(1): De operatie kost een extra cycle als er een page wordt overschreden.

ADC
Add memory to accu with carry
Flaggen: N, Z, C, V
Adressering: (1)
imm: OP:69NB:2NC:2
abs: OP:6dNB:3NC:4
zero: OP:65NB:2NC:3
(ind,x): OP:61NB:2NC:6
(ind),y: OP:71NB:2NC:5
zero,x: OP:75NB:2NC:4
abs,x: OP:7dNB:3NC:4
abs,y: OP:79NB:3NC:4
AND And memory with accu
Flaggen: N, Z
Adressering: (1)
imm: OP:29NB:2NC:2
abs: OP:2dNB:3NC:4
zero: OP:25NB:2NC:3
(ind,x): OP:21NB:2NC:6
(ind),y: OP:31NB:2NC:5
zero,x: OP:35NB:2NC:4
abs,x: OP:3dNB:3NC:4
abs,y: OP:39NB:3NC:4

ASL
Arithmetic shift left
Flaggen: N, Z, C
Adressering:
abs: OP:oeNB:3NC:6
zero: OP:06NB:2NC:5
accu: OP:oaNB:1NC:2
zero,x: OP:16NB:2NC:6
abs,x: OP:1eNB:3NC:7

BCC
Branch on carry clear
Flaggen: -
Adressering: (1)
rel: OP:90NB:2NC:3

BCS
Branch on carry set
Flaggen: -
Adressering: (1)
rel: OP:b0NB:2NC:3

BEQ
Branch on result equal
Flaggen: -
Adressering: (1)
rel: OP:f0NB:2NC:3

BIT
Test bits in memory with accu
Flaggen: N, Z, V
Adressering:
abs: OP:2cNB:3NC:4
zero: OP:24NB:2NC:3

BMI
Branch on result minus
Flaggen: -
Adressering: (1)
rel: OP:30NB:2NC:3

BNE
Branch on result not equal
Flaggen: -
Adressering: (1)
rel: OP:d0NB:2NC:3

BPL
Branch on result plus
Flaggen: -
Adressering: (1)
rel: OP:10NB:2NC:3

BRK
Break (forced interrupt)
Flaggen: I:1
Adressering:
impl: OP:00NB:1NC:7

BVC
Branch on overflow clear
Flaggen: -
Adressering: (1)
rel: OP:50NB:2NC:3

BVS
Branch on overflow set
Flaggen: -
Adressering: (1)
rel: OP:70NB:2NC:3

CLC
Clear carry
Flaggen: C:0
Adressering:
impl: OP:18NB:1NC:2

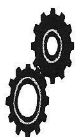
CLI
Clear interrupt disable status
Flaggen: I:0
Adressering:
impl: OP:58NB:1NC:2

CLV
Clear overflow flag
Flaggen: V:0
Adressering:
impl: OP:b8NB:1NC:2

CMP
Compare memory and accu
Flaggen: N, Z, C
Adressering: (1)
imm: OP:c9NB:2NC:2
abs: OP:cdNB:3NC:4
zero: OP:c5NB:2NC:3
(ind,x): OP:c1NB:2NC:6
(ind),y: OP:d1NB:2NC:5
zero,x: OP:d5NB:2NC:4
abs,x: OP:ddNB:3NC:4
bs,y: OP:d9NB:3NC:4

CPX
Compare memory and x-register
Flaggen: N, Z, C
Adressering:
imm: OP:e0NB:2NC:2
abs: OP:ecNB:3NC:4
zero: OP:e4NB:2NC:3

CPY
Compare memory and y-register
Flaggen: N, Z, C
Adressering:
imm: OP:c0NB:2NC:2
abs: OP:ccNB:3NC:4



zero: OP:c4NB:2NC:3

DEC
Decrement memory by 1
Flaggen: N, Z
Adressering:
abs: OP:ceNB:3NC:6
zero: OP:c6NB:2NC:5
zero,x: OP:d6NB:2NC:6
abs,x: OP:deNB:3NC:7

DEX
Decrement x-register by 1
Flaggen: N, Z
Adressering:
impl: OP:caNB:1NC:2

DEY Decrement y-register by 1
Flaggen: N, Z
Adressering:
impl: OP:b8NB:1NC:2

EOR Exclusive-or memory with accu
Flaggen: N, Z
Adressering: (1)
imm: OP:49NB:2NC:2
abs: OP:4dNB:3NC:4
zero: OP:45NB:2NC:3
(ind,x): OP:41NB:2NC:6
(ind),y: OP:51NB:2NC:5
zero,x: OP:55NB:2NC:4
abs,x: OP:5dNB:3NC:4
abs,y: OP:59NB:3NC:4

INC Increment memory by 1
Flaggen: N, Z
Adressering:
abs: OP:eeNB:3NC:6
zero: OP:e6NB:2NC:5
zero,x: OP:f6NB:2NC:6
abs,x: OP:feNB:3NC:7

INX Increment x-register by 1
Flaggen: N, Z
Adressering:
impl: OP:e8NB:1NC:2

INY Increment y-register by 1
Flaggen: N, Z
Adressering:
impl: OP:c8NB:1NC:2

JMP Jump to new location
Flaggen: -
Adressering:
abs: OP:4cNB:3NC:3
ind: OP:6cNB:3NC:5

JSR Jump to subroutine
Flaggen: -
Adressering:
abs: OP:4cNB:3NC:3

LDA Load accu
Flaggen: N, Z
Adressering: (1)
imm: OP:a9NB:2NC:2
abs: OP:adNB:3NC:4
zero: OP:a5NB:2NC:3
(ind,x): OP:a1NB:2NC:6
(ind),y: OP:b1NB:2NC:5
zero,x: OP:b5NB:2NC:4
abs,x: OP:bdNB:3NC:4
abs,y: OP:b9NB:3NC:4

LDX Load x-register
Flaggen: N, Z
Adressering: (1)
imm: OP:a2NB:2NC:2
abs: OP:aeNB:3NC:4
zero: OP:a6NB:2NC:3
abs,y: OP:beNB:3NC:4
zero,y: OP:b6NB:2NC:4

LDY Load y-register
Flaggen: N, Z
Adressering: (1)
imm: OP:a0NB:2NC:2
abs: OP:acNB:3NC:4
zero: OP:a4NB:2NC:3
zero,x: OP:b4NB:2NC:4
abs,x: OP:bcNB:3NC:4

LSR Logical shift right
Flaggen: N:0, Z, C
Adressering:
abs: OP:4eNB:3NC:6
zero: OP:46NB:2NC:5
accu: OP:4aNB:1NC:2
zero,x: OP:56NB:2NC:6
abs,x: OP:5eNB:3NC:7

NOP No operation
Flaggen: -
Adressering:
impl: OP:eaNB:1NC:2

ORA Or memory with accu
Flaggen: N, Z
Adressering:
imm: OP:09NB:2NC:2
abs: OP:0dNB:3NC:4
zero: OP:05NB:2NC:3
(ind,x): OP:01NB:2NC:6
(ind),y: OP:11NB:2NC:5
zero,x: OP:15NB:2NC:4
abs,x: OP:1dNB:3NC:4
abs,y: OP:19NB:3NC:4

PHA Push accu on stack
Flaggen: -
Adressering:
impl: OP:48NB:1NC:3

PHP Push flag-register on stack
Flaggen: -
Adressering:
impl: OP:08NB:1NC:3

PLA Pull accu from stack
Flaggen: N, Z
Adressering:
impl: OP:68NB:1NC:4

PLP Pull flag-register from stack
Flaggen: restored
Adressering:
impl: OP:28NB:1NC:4

ROL Rotate left
Flaggen: N, Z, C
Adressering:
abs: OP:2eNB:3NC:6
zero: OP:26NB:2NC:5
accu: OP:2aNB:1NC:2
zero,x: OP:36NB:2NC:6
abs,x: OP:3eNB:3NC:7

ROR Rotate right
Flaggen: N, Z, C
Adressering:
abs: OP:6eNB:3NC:6
zero: OP:66NB:2NC:5
accu: OP:6aNB:1NC:2
zero,x: OP:76NB:2NC:6
abs,x: OP:7eNB:3NC:7

RTI Return from interrupt
Flaggen: restored
Adressering:
impl: OP:40NB:1NC:6

RTS Return from subroutine
Flaggen: -
Adressering:
impl: OP:60NB:1NC:6

SBC Substract memory from accu with carry
Flaggen: N, Z, C, V
Adressering: (1)
imm: OP:e9NB:2NC:2
abs: OP:edNB:3NC:4
zero: OP:e5NB:2NC:3
(ind,x): OP:e1NB:2NC:6
(ind),y: OP:f1NB:2NC:5
zero,x: OP:f5NB:2NC:4
abs,x: OP:fdNB:3NC:4
abs,y: OP:f9NB:3NC:4

SEC Set carry-flag
Flaggen: C:1
Adressering:
impl: OP:38NB:1NC:2

SED Set Decimal-mode
Flaggen: D:1
Adressering:
impl: OP:f8NB:1NC:2

SEI Set Interrupt enable status
Flaggen: I:1
Adressering:
impl: OP:78NB:1NC:2

STA Store accu
Flaggen: -
Adressering:
abs: OP:8dNB:3NC:4
zero: OP:85NB:2NC:3
(ind,x): OP:81NB:2NC:6
(ind),y: OP:91NB:2NC:6
zero,x: OP:95NB:2NC:4
abs,x: OP:9dNB:3NC:5
abs,y: OP:99NB:3NC:5

STX Store x-register
Flaggen: -
Adressering:
abs: OP:8eNB:3NC:4
zero: OP:86NB:2NC:3
zero,y: OP:96NB:2NC:4

STY Store y-register
Flaggen: -
Adressering:
abs: OP:8cNB:3NC:4
zero: OP:84NB:2NC:3
zero,x: OP:94NB:2NC:4

TAX Transfer accu to x-register
Flaggen: N, Z
Adressering:
impl: OP:aaNB:1NC:2

TAY Transfer accu to y-register
Flaggen: N, Z
Adressering:
impl: OP:a8NB:1NC:2

TSX Transfer stack-pointer to x-register
Flaggen: N, Z
Adressering:
impl: OP:baNB:1NC:2



TXA Transfer x-register to accu
Flaggen: N, Z
Adressering:
impl: OP:8aNB:1NC:2

TXS Transfer x-register to stack-pointer
Flaggen: -
Adressering:
impl: OP:9aNB:1NC:2

TYA Transfer y-register to accu
Flaggen: N, Z
Adressering:
impl: OP:98NB:1NC:2

Aangepaste printroutine

De video chip van de Commodore 64 maakt het mogelijk om de standaard karakterset te veranderen. Met wat creativiteit kunt u dus zelf een sierlijk lettertype ontwerpen en deze in de computer invoeren m.b.v. een karaktereditor. Alle teksten, zoals listings, worden dan in dit lettertype op het scherm afgedrukt. Tot zover verloopt alles vlekkeloos. Op het moment dat u echter een tekst naar de printer stuurt, komt u in de problemen. De tekst wordt namelijk in het normale lettertype afgedrukt en niet in het prachtige lettertype, dat u zojuist met veel moeite ontworpen heeft. De printer biedt namelijk niet de mogelijkheid om een zelf-ontworpen karakterset af te drukken, maar drukt alle tekst in gewone drukletters af. De wat duurdere printers kunnen teksten wel in verschillende lettertypes, zoals Bold en Italics, afdrukken. Deze lettertypes zijn echter door de fabrikant gedefinieerd en kunnen niet veranderd worden en dat is juist wat u wilt. We zullen u in deze aflevering een idee aan de hand doen waarmee u het zojuist beschreven probleem kunt oplossen. De meeste printers beschikken over een zogenaamde high resolution print mode. Deze mode is een vereiste voor het afdrukken van nieuwe lettertypes. Als uw printer hierover niet beschikt, dan is het afdrukken van uw eigen karakterset helaas onmogelijk. Het idee komt op het volgende neer: We passen de routine die een karakter afdrukt zodanig aan, dat elk karakter dat naar de printer wordt gestuurd, wordt behandeld door een zelf-geschreven routine en elk karakter dat naar een ander apparaat, zoals scherm of diskdrive, wordt gestuurd, normaal wordt afgehandeld. De zelf-geschreven routine moet er dan voor zorgen dat het naar de printer gestuurde karakter pixel voor pixel wordt afgedrukt in high resolution mode. De routine moet hiervoor gebruik maken van het bitpatroon van het karakter, dat in het geheugen van de computer ligt opgeslagen. We zullen nu laten zien hoe de routine die een karakter afdrukt, aangepast kan worden. Deze routine wordt geactiveerd met een sprong naar adres $FFD2. Dit adres zal u, als machinetaal programmeur, ongetwijfeld bekend voorkomen. (Bij aanvang van de routine bevat de accumulator de ascii code van het karakter dat afgdrukt moet worden.) De computer springt vervolgens indirect via de vector op adres $0326 naar adres $F1CA. Het enige dat we nu moeten doen om de routine aan te passen, is het veranderen van de vector op $0326. Deze vector moet naar onze eigen routine gaan wijzen. Het volgende programma geeft een kort voorbeeld. Het programma drukt na elk karakter dat naar de printer gestuurd wordt, een spatie af.

```
lda # art ;  Verander vector
sta $0326
lda #start
sta $0327
rts start
pha ;    Bewaar accu
lda $9a  ;Karakter naar
cmp #$04 ;printer ge-
stuurd?
beq printer
pla  ;Karakter niet
jmp $f1ca    ;naar prin-
ter! printer
pla  ;Karakter wel
cmp #$0d ;naar printer!
beq eoln ;Einde regel?
jsr $f1ca    ;Druk karak-
ter af
pha
lda #$20
jsr $f1ca    ;Druk spatie
af
pla
rts eoln
jmp $f1ca    ;Einde regel
```

De routine 'start' kijkt in adres $9A of het karakter naar de printer wordt gestuurd. Dit adres bevat het nummer van het apparaat waarnaar het karakter wordt gestuurd. Als het karakter niet naar de printer wordt gestuurd, wordt er naar adres $F1CA gesprongen; het karakter wordt dan op de normale manier afgehandeld en afgedrukt. De routine 'printer' drukt na elk karakter een spatie af. De routine kijkt eerst of het karakter een 'return' (ASCII 13) is. Als dit het geval is, dan moet deze naar de printer worden gestuurd en moet er geen spatie worden afgedrukt. Anders wordt eerst het karakter afgedrukt en vervolgens een spatie (ASCII 32). Op gelijke wijze kunt u nu een routine schrijven die teksten in zelf-gedefinieerde lettertypes afdrukt. Daarvoor moet u de routine 'printer' vervangen door een routine die elk karakter in high resolution mode afdrukt. Bij sommige printers is het noodzakelijk dat u van tevoren opgeeft hoeveel gegevens u in high resolution mode verstuurt. U kunt dan bijvoorbeeld eerst alle karakters in een buffer in de computer opslaan en pas gaan printen op het moment dat een 'return' gestuurd wordt. U weet dan immers hoeveel karakters er gestuurd worden. Veel succes en plezier bij het schrijven van uw routine.

Save en replace

Officieel bestaat er een mogelijkheid om een programma op disk te overschrijven met een ander programma met dezelfde naam. Dit kan heel handig zijn als u bijvoorbeeld bezig bent met programmeren. U kunt dan een nieuwe versie van het programma over de oude versie heen saven. Voorwaarde is dan wel dat de oude versie en de nieuwe versie dezelfde naam moeten hebben. Het commando voor zo'n replace is het volgende:
save"‘0:progr.naam",8

Voor de programma naam moet u dus eerst een at-sign (apestaartje) en een nul typen. In de praktijk blijkt echter dat dit commando vaak niet werkt. Sterker nog, vaak worden er ook nog andere programma's die op dezelfde disk staan, volledig de vernieling in geholpen. Dit ligt gewoon aan het feit dat dit commando niet goed geprogrammeerd is in het ROM. We raden u dan ook aan deze optie NOOIT te gebruiken! U kunt de replace-optie natuurlijk wel nabootsen, door de nieuwe versie eerst onder een andere naam te saven en vervolgens de oude versie te scratchen. Daarna kunt u de nieuwe versie nog eventueel renamen. U kunt dit op de volgende wijze doen:

save"nieuwe versie",8 open 1,8,15 print#1,"s0:oude versie"
(* scratch oude versie *)
print#1,"r0:oude versie=nieuwe versie"
(* rename nieuwe versie *)
close 1

U moet bij het renamen er wel op letten, dat u na de = de naam zet van het programma dat u wilt veranderen, en voor de = de nieuwe naam die het programma moet krijgen (feitelijk een onlogische volgorde).

Rekenroutines

Het rekenen in machinetaal is vaak een lastig karwei om te programmeren. Hoewel het altijd vervelend programmeren blijft, kunt u zich een hoop werk besparen door de ingebouwde ROM-routines te gebruiken. We zullen hier een overzicht geven van de



Rekenkundige bewerkingen		
bewerking	routine	adres
z = x + y	add	b867
z = x * y	multiply	ba28
z = x - y	substract	b853
z = x / y	divide	bb0c
z = x ^ y	power	bf7b
z = x and y	and	afe9
z = x or y	or	afe6

Figuur 1

belangrijkste rekenroutines, die de Commodore 64 bezit. Daarbij gaan we ervan uit dat de lezer al weet hoe het rekenen in de praktijk gebeurt, en dus al bekend is met de floating point accumulator (FPAC). We zullen hier dan ook niet uit gaan leggen hoe u met de FPAC moet werken.

In **figuur 1** staat een lijst van rekenroutines die u kunt gebruiken. De Y moet steeds in de FPAC staan. De X moet u meegeven en wel zo: Het lo-byte van het adres waar zich getal X bevindt. moet u van te voren in de accu plaatsen, en het hi-byte in het y-register. Het resultaat Z komt weer in de FPAC te staan. **Figuur 2** bevat een lijst functies. De X moet hier steeds in de FPAC staan. Het resultaat Y komt na aanroepen van de routine in de FPAC te staan.

Hylke Sprangers & Michel de Boer

Functies		
bewerking	routine	adres
y = sqr(x)	sqr	bf71
y = sin(x)	sin	e26b
y = cos(x)	cos	e264
y = tan(x)	tan	e2b4
y = atn(x)	atn	e30e
y = log(x)	log	b9ea
y = exp(x)	exp	bfed
y = int(x)	int	bccc
y = sgn(x)	sgn	bc39
y = abs(x)	abs	bc58
y = -x	neg	bfb4

Figuur2

14e Computer Info Beurs

3 november in de RAI

10.00-17.00

Toegang: *f* 7,50



GEOS INSTALLEREN, EEN PROBLEEM?

Na aanschaf van een GEOS 2.0 versie moet een installatieprocedure worden door geworsteld. Ik zal proberen wat licht in het duister te scheppen.



Nadat u de doos heeft opengemaakt zult u vier diskettes en een boek vinden. Als een goed computergebruiker weet u dat u niet zonder handleiding kunt en zult u ook eerst de handleiding doorbladeren. Hier vindt u in het Hoofdstuk Getting Started de paragraaf Installing GEOS. Als ik dit zo lees dan kan ik mij best voorstellen dat het u duizelt voor de ogen. Ik zal proberen of ik het stapsgewijs kan uitleggen.

Het booten van GEOS

° Sluit uw Commodore, disk drive(s), monitor en uw printer aan volgens de gebruikelijke instrukties. Heeft u een RAM-uitbreiding; sluit deze dan voorzichtig aan in de cartridgepoort van uw Commodore.

° Doe uw monitor, computer en diskdrive aan.

N.B.:
Gebruik tijdens het installeren van GEOS 2.0. maar 1 diskdrive. Een tweede diskdrive die aanstaat kan de installatieprocedure verstoren en daardoor uw GEOS 2.0. diskettes beschadigen.

° Doe de GEOS SYSTEM disk (dit is disk nummer 1) in de diskdrive en sluit de deur.

° Het scherm meldt zich met het Commodore 64 Basic scherm. Typ Load"GEOS",8,1 en druk op RETURN. Vervolgens zullen de boodschappen "Searching for GEOS", "Loading" en "Booting GEOS" op uw scherm te zien zijn.

° Een dialoogvenster zal u vragen: "Please insert disk Back-up System". U neemt nu de SYSTEMdiskette (disk 1) uit de diskdrive en doet de BACK-UP diskette(disk 2), in de diskdrive. U sluit de drive en drukt op <RETURN>.

° Een ander dialoogvenster zal u nu vragen of u de GEOS SYSTEMdisk d.m.v. een serienummer wilt "locken" aan uw reeds eerder geïnstalleerde applikaties. Als u in het bezit bent van bijv. DeskPack, GeoFile, GeoCalc of andere GEOS applikaties antwoordt dan met ja door het pijltje op "YES" te zetten. Als u geen GEOS applikaties heeft dan alleen deze GEOS 2.0. antwoord dan met "NO".

Het pijltje kunt u ook over het scherm bewegen door de CRSR-toetsen te gebruiken.

° Als u het pijltje op YES of NO zet druk dan op <RETURN>.

° Heeft u YES gekozen ga dan verder met stap 8. Heeft u op NO gedrukt ga dan verder met stap 9.

° Bij YES zal een dialoogvenster u vragen om een applikatie in de drive te doen die al reeds geïnstalleerd was op uw oude BOOTdisk bijv. DeskPack, GeoFile of GeoCalc. Hier-



mee krijgt u de mogelijkheid om de oude applikatie te locken met de nieuwe BOOTdisk. U gaat als volgt te werk: U haalt de BACK-UP disk (disk 2) uit de drive en doet er een originele GEOS applikatiedisk in. U sluit de drive en drukt op <RETURN>.

° Het volgende dialoogvenster zal u vragen om de SYSTEMdisk (disk 1). Haal de disk die in de drive zit (disk 2) eruit en doe de SYSTEMdisk (disk 1) erin. Druk nu op <RETURN> om verder te gaan.

° Een ander dialoogvenster vraagt u vervolgens om de BACKUPdisk (disk 2). Verwijder de SYSTEMdisk (disk 1) uit de drive en doe de BACK-UPdisk (disk 2) er vervolgens in en druk weer op <RETURN>.

° Het dialoogvenster wat nu verschijnt op uw scherm vraagt nu weer om de SYSTEMdisk (disk 1). Verwissel de BACK-UP disk (disk 2) voor de SYSTEMdisk (disk 1)en druk op <RETURN>. Na enkele momenten zult u de deskTop op het scherm zien verschijnen.

Installeer uw applikaties

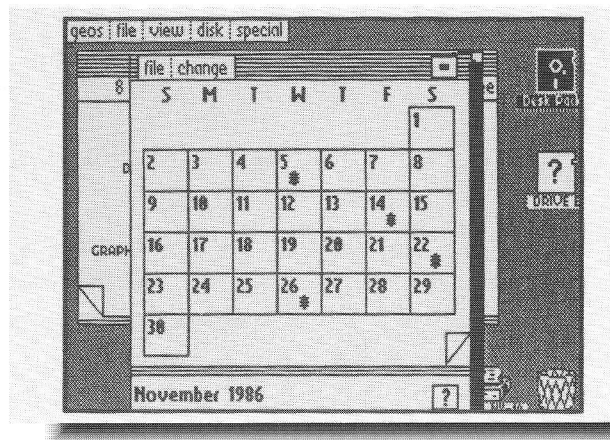
De applikaties die u zult moeten installeren zijn GeoWrite, GeoMerge en GeoSpell. U kunt de applikaties niet eerder gebruiken dan dat deze door u geïnstalleerd zijn.

Hoe installeert u de applikaties

° Sluit de SYSTEMdiskette door met uw pijltje op het CLOSE (sluit) icon te gaan staan dat zich in de rechterbovenhoek bevindt van het werkblad.Of gebruik het toetsenbord kommando C= C.(houdt de C= toets ingedrukt en toets de letter C.)

° Als het pijltje op CLOSE icon staat drukt u op de (vuur)-knop (dit noemt men ook aanklikken). Zodra u het CLOSE icon aanklikt heeft u de diskette 'gesloten'.

° Haal de SYSTEMdisk (disk 1) uit de diskdrive en doe de APPLICATIONS disk (disk 2, kant B) in de drive. Klikt u nu het DRIVE icon aan van de diskdrive waarin u de diskette APPLICATIONS heeft gedaan (bijv. drive A).



° De directory van de "APPLICATIONS"disk zal nu op het scherm verschijnen. Ga met het pijltje naar het icon met de naam GeoWrite en klik deze 2 x snel aan (dit wordt de dubbelklik methode genoemd). Het scherm zal leeggemaakt worden en een nieuw venster verschijnt met de mededeling "GeoWrite installed", dit betekent: GeoWrite is geïnstalleerd (heeft het serienummer van uw GEOS 2.0 disk overgenomen). Om nu terug te keren naar de deskTop, klikt u op OK of <RETURN>.

° Sluit nu de "APPLICATIONS" disk door het CLOSE-icon rechtsboven aan het werkblad aan te klikken. Of gebruik C= C.

° Verwijder de "APPLICATIONS" disk uit de diskdrive en plaats de disk met de titel "Write Utilities" (disk 3, kant A) in de diskdrive.

° Klik het drive A icon aan (hier zal een vraagteken in staan), en de directory van de "Write Utilities" disk zal op het werkblad verschijnen.

° Dubbelklik het GEOMERGE icon aan. Ook nu zal weer een dialoogvenster verschijnen met de mededeling "GEOMERGE installed", wat betekent dat GeoMerge geïnstalleerd en klaar is voor gebruik. Klik op OK of <RETURN> om terug te keren naar de deskTop.

° Herhaal de installatie procedure uit punt 8 met de GeoSpell disk (disk 3, kant B) en installeer GeoSpell.

N.B.: Als u op het scherm de mededeling "Cannot install on this disk" krijgt, overtuig u er dan van dat de juist disk in de drive is geplaatst. U zult een applikatie altijd eerst moeten installeren voordat u deze kopieert naar een werkdisk. Tijdens het installeren van een applikatie wordt namelijk het serienummer van uw syteem aan de applikatie toegevoegd.

U heeft nu GEOS en de GEOS Applikaties geïnstalleerd.
In grote lijnen heeft deze installatie geleid tot plaatsing van een uniek serienummer voor beide diskettes. U kunt dan ook met beide diskettes GEOS opstarten. Beter is om de Systemdisk te gebruiken voor het opstarten en de Back-up disk op te bergen voor noodgevallen. Mocht onverhoopt uw Systemdisk stuk gaan, kunt u altijd met de Back-up disk doorgaan. De stukkende disk kunt u opsturen naar de St. GEOS Gebruikers voor reparatie.

Serienummers

De eerste keer dat u de System Disk boot, zal het programma u vertellen om afwisselend de System- en de Back-up disk te gebruiken. Zodra de installatie is voltooid zullen beide diskette hetzelfde serienummer bevatten. Het serienummer is een identifikatie op de diskette dat zich verbindt met de GEOS Bootdisk en de GEOS applikaties die u aanschaft aanpast. Als u een door u aangeschafte applikatie installeerd krijgt deze automatisch hetzelfde serienummer als uw bootdisk. De melding "... applikation installed" verschijnt alleen eenmalig.

Booting Problemen

° Als uw GEOS systeem niet wil "booten" zijn er een aantal reden waarom dat gebeurd. Een oplossing is er dan meestal wel.

° Probeer met de Back-up Disk te booten. Beide diskettes zijn bootable versies van GEOS. Als één diskette wel boot en de andere niet dan heeft u waarschijnlijk een defekte disk. In dit geval kunt u deze opsturen naar de St. GEOS Gebruikers en hem ter reparatie aanbieden. Dit ge-



schiedt kosteloos. Als geen van beide diskettes booten dan is er een mogelijkheid dat u hardware problemen heeft.

° Overtuig u ervan dat alle hardware op de juiste manier is aangesloten. Hardware die niet geheel compatible met elkaar is kan een goede werking van GEOS verstoren. Door alle niet voor GEOS benodigde extra's kunt u het beste verwijderen.

° Snellaadcartridges zoals Final en Powercartidge hebben weleens aangetoont erg dominant te zijn en een goede werking van GEOS te verstoren. Ook het uitzetten van de printer helpt bij het booten van GEOS. De printer kunt u dan aanzetten als de deskTop op het scherm staat.

° Probeer uw GEOS diskettes ook eens bij iemand anders te booten. Hiermee kunt u dan al heel snel zien of het probleem in de diskettes of in uw harware zit.

° De melding "Please Reboot Your System with the disk that was first used to run......"(applikatienaam) heeft een heel eigen probleem. Als u deze melding tegenkomt probeert u een reeds geinstalleerde applikatie op uw systeem te starten. Deze applikatie zal naar alle waarschijnlijkheid niet uw eigendom zijn. De verkregen applikatie is op een ander GEOS systeem voorzien van een serienummer dat niet overeenkomt met uw serienummer. Bent u in het bezit van de originele applikatie diskette dan kunt u deze opsturen naar de St. GEOS Gebruikers.

Zij kunnen de desbetreffende applikatie voor u herstellen, zodat u weer volledig gebruik kunt maken van de applikatie.
LET WEL: alleen alsd u in bezit bent van de originele disk.

° Met de melding"file not found" kan het zijn dat:
° u het klepje van de diskdrive niet heeft gesloten;
° u een verkeerde disk in de drive heeft geplaatst;
° uw diskdrive lees/schrijf kop geinitialiseerd moet worden.

GEOS DISKETTES MOGEN NOOIT WORDEN AFGEPLAKT.

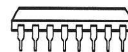
Waarom niet?

Door de opbouw van de GEOS VLIR bestanden en de intelligentie van de GEOS Kernal heeft GEOS de mogelijkheid om zelf ruimte te gebruiken waar en wanneer het systeem het blieft. Ongelooflijk?
Het is het beste uit te leggen door middel van een voorbeeld.
We gaan er van uit dat u twee diskdrives heeft. In drive A zit de BOOTdisk en drive B bevat bijvoorbeeld uw geoWrite werkdisk.
U bent bezig in geoWrite met een dokument. U maakt gebruik van de Text- en Photo Manager en van de het Notepad en de calculator.
Tevens heeft u nog bijv. 18 KB free op de werkdisk. Genoeg voor een brief denkt U.
Nu de werkelijkheid. Als u geoWrite opstart opent GEOS voor U geoWrite maar tegelijkertijd opent GEOS op uw werkdisk een zogenaamde scrap-file waarin de informatie door de kernal wordt opgeborgen over waar u bent en wat u gebruikt en/of gaat gebruiken. Er wordt namelijk gebruik gemaakt van de zogenaamde overlay techniek. Deze overlay techniek is goed te zien als u gebruik maakt van een desk-accessoire als bijvoorbeeld de calculator. geoWrite blijft gewoon op het scherm staat en de calculator wordt daar overheen geprojekteerd. Om het systeem nu te laten weten dat als u klaar bent met de calculator terugkeert naar geoWrite en niet naar de deskTop wordt deze informatie opgeslagen in deze scrap-file. Hoe meer u gebruik maakt van desk-accessoires en of bijv Fonts des te groter wordt deze scrapfile.
Het zal niet de eerste keer zijn dat een scrap-file de 25 KB overschrijdt. Om terug te keren naar ons voorbeeld waar u met het nodige gebruik van Fonts, berekeningen, photo- en text-scraps u brief heeft gemaakt. en klaar bent voor het printen.
U wilt vanaf de deskTop printen en u kiest in het File menu voor Quit. U keert terug naar de deskTop en u kunt printen. U heeft mischien nog 12 KB free op de disk en er is niets aan de hand.
De kernal heeft voordat u terugkeert naar de deskTop de scrap-file gewist. U heeft het rode lichtje van Drive A wel een paar keer zien oplichten maar daar verder geen aandacht aan geschonken. Er was ook trouwens niets gebeurd. U had de disk ook niet afgeplakt met een write-protect sticker.

Systeem error near $....

Zou u de BOOTdisk wel afgeplakt hebben dan gebeuren er hele nare dingen. Op het moment dat de kernal constateerd dat de scrap-file alle vrije ruimte op disk nodig heeft zal de Kernal op zoek gaan naar vrije ruimte op de disk in de andere drive. Vindt hij deze dan is er niets aan de hand. Dit verklaart alleen maar het oplichten van het rode ledje van drive A. Maar vindt de kernal de vrije ruimte op de disk in drive a maar kan deze niet beschrijven dan zal de Kernal de gereserveerde ruimte voor uw brief gaan gebruiken. Dit zal resulteren in het niet kunnen bewaren van uw brief. Ook al had u nog 18 KB free op de disk voordat u met geoWrite begon, toch zal u de melding disk full op uw scherm krijgen. Tevens kan het resulteren in een Systeem error near....
Dit betekend een totale crash van het systeem. Ook zult u uw brief opnieuw moeten maken. De enige remedie tegen dit probleem is: GEOS diskettes niet afplakken......

Pim Broekhuijzen



DE ENHANCED CHIP SET

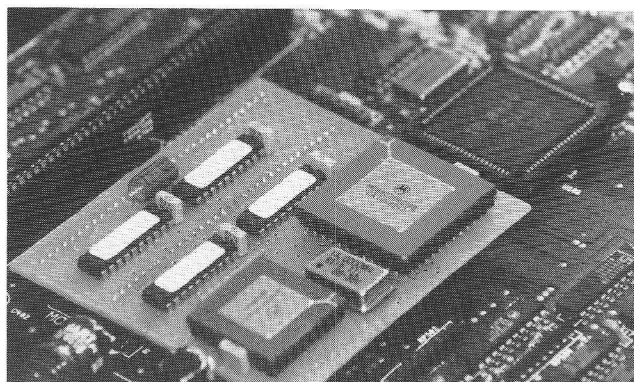
Meer coprocessor-kracht op de Amiga

Met de Amiga 3000 heeft Commodore die nieuwe ECS-coprocessors geïntroduceerd. Deze Enhanced Chip Set verbetert de prestaties van Agnus Paula en Denise aanzienlijk. Beeldkwaliteit, multitasking-faciliteiten en video-RAM-beheer zijn nu weer helemaal up-to-date. Hoewel de ECS vooral op de multimediamarkt gericht is komt zij ook goed van pas bij andere applicaties en het upgraden van oudere Amiga-modellen.

De Amiga-PC-lijn staat bekend om haar uitbestedingswerk. De "onderaannemers" Fat Agnus, Denise en Paula nemen de Motorola 680X0 CPU veel werk uit handen. Daardoor wordt er flink op de microprocessortijd bespaard en lopen de video- en multitasking- prestaties op de Amiga aanmerkelijk soepeler en sneller dan op conventionele PC-systemen. Het is niet voor niets dat andere multimediasystemen zoals IBM's Audio Visual Connection (AV) ook coprocessorkaarten gebruiken! Er bestaat een belangrijk verschil tussen het Amiga-chiptrio en PC's met coprocessor-uitbreidingskaarten. Agnus, Paula en Denise zijn van meet af aan geintegreerd geweest met de Amiga- hoofdkaart. Bij andere PC-systemen was en is er doorgaans sprake van uitbreiding via een busslot. Topologisch en communicatie-technisch gezien levert de Amiga met zijn geïntegreerde chipset betere prestaties dan optionele coprocessors.

Chip-evolutie

Denise, Agnus en Paula hebben in de loop van de vijf Amiga-jaren de nodige veranderingen ondergaan. Niet echt revolutionair, maar wel een constante verbetering. Een belangrijke verbetering was bijvoorbeeld verhoging van het Chip-RAM-bereik van 512 kB tot 1 MB. Multitasking-, geluids- en videotoepassingen werden daardoor flink verbeterd. Ook nam de geboden beeldkwaliteit en het kleurenpalet per nieuwe Amiga-telg toe.
Bij de introductie van de multimedia-Amiga 3000 werden weer een aantal verbeteringen doorgevoerd. Geen echte knallers, maar toch weer net die finishing touch die het systeem up-todate afmaakt. De nieuwe ECS bestaat uit Super Agnus (ECS-Agnus, of gereviseerde Fat Agnus), Super Denise en een nieuwe video-IC. Paula werd nauwelijks gewijzigd.

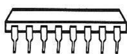


Amiga 2000/2500

Vanaf de herfst in 1989 worden al Amiga 2000/2500-modellen geleverd met een 1 MB Fat Agnus-chip. Een grote 1 Meg Chip RAM- sticker op de doos vertelt de eigenaar of de dikke Super Agnus aan boord is of niet. Twijfelt u dan brengt het AVAIL-commando uitkomst. In de kolom Maximum prijkt in de eerste rij als derde item (= marked chip) het getal 1040152. Zo niet dan zit er dus geen 1 MB Fat Agnus in de Amiga.
Inbouw kan de gevorderde hobbyist zelf doen. Er komt wel enig trek-, wrik-, kraben soldeerwerk bij te pas. De minder geschoolde Amiga-bezitter wende zich in deze beter tot een geautoriseerde dealer of Commodore Nederland.
Bij de Amiga 500 zijn er nog enkele 512 kB Agnus-modellen in omloop.
Het gros is echter voorzien van de nieuwe "dikke". Een probleem is dat de Amiga 500 standaard met 512 kB geleverd wordt. U kunt dus niet direct zien of Agnus met 1 MB aan Chip-RAM kan omgaan en er zijn modificaties nodig om bij uitbreiding de 500 op 1 MB aan Chip-RAM over te schakelen! De nieuwe ECS Denise werkt alleen met de Super Fat Agnus. In een machine met de 1 MB Chip zit binnenkort ook de ECS Denise. Heeft u nog geen Super Denise dan is het zelf vervangen van deze chip een fluitje van een cent. Gewoon een chiptrekker en -plugger hanteren.

De 1 MB Fat Agnus

Dikke Agnus kwam afgelopen herfst op de markt en kan zoals de naam reeds aangeeft 1 MB aan Chip-RAM in één keer adresseren. Voorheen was 512 kB aan vrij CHIP-RAM voor geavanceerde video- en audiotoepassingen eigenlijk veel te krap. Voor een snelle vloeiende weergave was veel kunst-en-vliegwerk nodig.
In feite doet de nieuwe Fat Agnus niets anders dan een nieuwe geheugenverdeling maken. Er is dus geen sprake van uitbreiding van het totaal te installeren RAM-geheugen. De coprocessors kunnen nu gewoon dubbel zoveel Chip-RAM uit het beschikbare RAM aanwenden. Op een 1 MB Amiga in principe dus één MegaByte, maar dan blijft er voor de CPU niets over. Gelukkig valt dit in de praktijk wel mee. Het blijft echter raadzaam om bij grote grafische toepassingen de Amiga met minimaal 2 MB uit te rusten.



Bij de Amiga 3000 kan maximaal 2 MB aan vrij Chip-RAM door de ECS Agnus aangewend worden.
Tot de extra mogelijkheden van de nieuwe Agnus behoren o.a.:

- Verbeterde multitasking bij grafisch intensieve software, bijvoorbeeld twee HIRES-paintpakketten;
- Verhoging van het HIRES-beeld, bijvoorbeeld 1000 x 1000; pixels in 16 kleuren
- Het starten van een grafisch intensief programma vanuit de Workbench zonder eerst het systeem te behoeven resetten;
- Verlenging van de afspeelduur (payback) bij animaties en muziek.

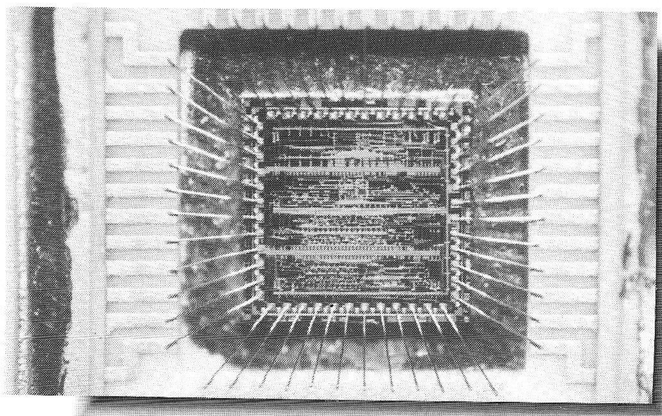
Ook de Blitter-component onderging een belangrijke verbetering.
De Blitter houdt zich bezig met het animeren van objecten of sprites.
Voorheen ging dit in 1 K x 1 K-porties. Met de nieuwe Agnus kan de Blitter 32 K x 32 K grafische brokken aan. Grote animatie-objecten en scrollen- en de schermen zijn nu makkelijker te realiseren.
Video-makers ergerden zich al jaren wild aan het Amerikaanse NTSC-systeem dat roet in het Europese PAL-eten gooit. Allerlei lastige en vooral dure hardware-aanpassingen waren nodig om de Amiga PAL-compatibel te maken. Met de nieuwe Agnus is een goede PAL-display geen probleem meer. Met Amiga DOS 2.0 kan er zelfs softwarematig uit de Workbench tussen PAL en NTSC gekozen worden.

Super Denise

Sinds kort is ook de Super Denise leverbaar. Super Denise is sterk afhankelijk van de nieuwe Fat Agnus. Zij werkt zelfs niet zonder de 1 MB "dikke". Op videogebied zet Denise nu twee maal zoveel pixels op dezelfde beeldlijn neer als voorheen. De pixels zijn dan wel half zo groot als normaal, hetgeen o.a. aan de muiscursor zichtbaar is.
Met deze Super-HIRES display bereikt ECS Denise een non-interlaced resolutie van 1280 x 200 pixels of een interlaced HIRES-scherm met 1280 x 400 pixels. Wel wordt het kleurenpalet door deze HIRES-manoeuvre beperkt tot maximaal 4 kleuren uit 64. Een andere monitor of genlock is gelukkig niet nodig.
Bezitters van VGA-monitors kunnen dank zij de ECS Denise hun beeldbuis nu direkt op de Amiga aansluiten. Voorheen was daar een prijzige optionele videokaart zoals bijvoorbeeld de Flickerfixer voor nodig. Denise biedt softwarematige besturing van de beeldsynchronisatie- en blanking-signalen en kan via doublescan (verdubbelde horizontale scanrate) de Amiga-resoluties naar een 31,5 KHz VGA-monitor vertalen. Met de ECS Denise is het in principe mogelijk om de Amiga aan allerlei displays en lastige video- opnamesituaties (bijvoorbeeld het filmen van een monitor) aan te passen.
Op een VGA-monitor krijgt u nu 640 x 480 pixels, non-flickering in 4 kleuren uit 64. Een hele verbetering tijdens de Werkbank- display (2.0 versie) en het gebruik van DTP- of CAD-software.
Tot de door ECS Denise geboden nieuwe videoresoluties behoren:

- Productivity (Workbench 2.0!) met 640 x 480 pixels non- interlaced;
- Interlaced Productivity met 640 x 960 pixels;
- Super-HIRES non-interlaced met 1280 x 200 pixels;
- Interlaced Super-HIRES met 1280 x 400 pixels.

Al deze grafische modi zijn gelimiteerd tot 4 kleuren uit 64 en twee bitplanes.
De desktopvideo-er wordt verblijd met een verbeterde genlock- besturing.
Bij de oude Denise kon de genlock slechts de achtergrondkleur en het kader om het beeld beïnvloeden. Met de nieuwe ECS ColorKey is elke denkbare combinatie van paletkleuren transparant te maken. Met de BitPlane-Key wordt in plaats van een kleur of kleurcombinatie een bitplane als transparant masker gebruikt. U kunt dus elke vorm als transparant masker gebruiken voor boeiende video-effecten.
Kortom ECS Denise biedt alles wat het multimedia-displayhartje begeert.



Display-enhancer

Behalve de ECS-Denise is er ook sprake van een een Display- enhancer. Het gaat hier om een IC die de taken van de optionele Flickerfixer (Microwave) overneemt. Net als de Flickerfixer neemt de ECS Display-enhancer de hinderlijke interlace-flickering weg door een zelfde resolutie in maximaal 4096 HAM-kleuren te ont-interlacen. Denise maakt gewoon gebruik van deze IC bij het verbeteren van de conventionele Amiga-display en biedt daarnaast nog de eigen nieuwe HIRES 4-kleuren grafische modes.
Let even op het volgende belangrijke verschil. De Flickerfixer maakt de standaard Amiga-display geschikt voor VGA-monitors en neemt de hinderlijke flikkering weg. De ECS Denise is van huis uit al geschikt voor non-flickering VGA in de nieuwe grafische modi. Een doublescan over de Flickerfixer weergeven gaat daarom falikant mis.

Beschouwing

De nieuwe ECS-uitbreiding van de Amiga-lijn is een belangrijke stap voorwaarts. Vooralsnog mikt Commodore hiermee voornamelijk op multimedia-toepassingen.
Toch komt het verbeterde chiptrio ook bij de gewone software-display, DTP, CAD(D) en muziek goed van pas.
Van een ware revolutie is geen sprake. De aangebrachte verbeteringen moeten gezien worden als een update van een beproefd concept naar de huidige video- en displaynormen. Er is natuurlijk nog veel meer mogelijk als we naar het neusje van de zalm bij Super VGA- en 8514-adapters kijken. Gezien de prijs biedt de ECS echter veel waar voor zijn geld.
Wie meer wil zal heel diep in de beurs moeten tasten.
U.S

Checksum C-64

Syntax Checksum

Het overtikken van een listing kan een heel karwei zijn en als u een beetje normaal mens bent dan maakt u daarin beslist een aantal fouten. Nu is niets moeilijker om de fouten uit je eigen werk te halen. Al geruime tijd heeft Jan Bodzinga hiervoor een zgn. Checksum-programma geschreven. Om de vele nieuwe lezers van Commodore-info te helpen volgt hieronder nog een keer een volledige uitleg over de werking van dit programma, waarmee het, hoe vreemd dat misschien ook lijkt, echt mogelijk is om met behulp van dit programma de fouten in elke door ons geplaatste listing op te sporen.
Hiervoor gaat u als volgt te werk:
1. U tikt de listing heel zorgvuldig over en SAVEt hem voordat u het programma RUNt op een diskette of cassette.
2. U tikt het RUN commando in. Mocht het programma de boodschap 'FOUT in dataregels!' geven dan heeft u een fout bij het overtikken gemaakt. Herstel dan de fout en SAVE de verbeterde versie. Mocht het programma met de boodschap 'data is weggezet checksum testen met sys...'komen dan is tot dusver alles goed. Het programma is nu in een stukje machinetaalgeheugen gezet. Als u het NEW commando geeft blijft het toch in de computer staan.
Alle door ons geplaatste programma's zijn in Basic geschreven.
Als u een programma heeft overgetikt SAVE het eerst, mocht er iets mis gaan dan hoeft u niet de gehele listing opnieuw te gaan intikken. Als u nu een programma op fouten wilt gaan controleren dan kunt u dat in het geheugen laden(wel eerst het checksum programma hebben gerund). Vervolgens typt u zonder het programma te runnen de opdracht sys 49152(c-64) of sys 1536 (c-16 en plus/4)in.
Als alles goed is gegaan loopt er nu een rij regelnummers over het scherm met getallen erachter. Dezelfde lijst staat ook achter elk door ons geplaatste programma. Wijkt nu een nummer achter een regelnummer af van het nummer dat in het blad staat dan heeft u in die regel iets anders ingetikt dan er in het blad stond. U kunt de stroom getallen d.m.v. de RUN/STOP toets pauzeren en weer vervolgen met de F1 of F7 toets.
Het is uitermate belangrijk dat u goed met dit programma overweg kunt en mocht u het niet goed werkend krijgen bel dan gerust even met onze listingservice telefoonlijn. (Maandag 17.00 - 21.00 uur. Telefoonnummer 02155-25162.)
De laatste tijd wordt er weer veel gebeld zodat U nogwel eens in gesprek krijgt, daarom houdt uw vraag kort, vermeld in welk blad het desbetreffende artikel stond. Heeft U veel vragen, of zis uw vraag erg uitgebreid, doe het dan schriftelijk, zodat we veel mensen op de maandag avond te woord kunnen staan.

```
  1 rem *****************************
  2 rem basic loader "SYNTAX.CHECKSUM"
  3 rem na de commando's "run" en "new"
  4 rem blijft dit programma in het ge-
  5 rem heugen. laad het te testen pro-
  6 rem gramma en tik daarna sys 49152.
  7 rem *****************************
10   i=49152   :rem beginadres
20   reada:ifa<0then40:rem data ingelezen
30   pokei,a:i=i+1:b=b+a:goto20
40   if b<>16844thenprint"[SHIFT-CLR]fo
     ut[SPACE]in[SPACE]dataregels!":b=0:end
50   poke49184,148:poke49185,192
55   i=49300
60   read a: ifa<0then80
70   pokei,a:b=a+b:i=i+1:goto60
80   if b<>20068thenprint"[SHIFT-CLR]fo
     ut[SPACE]in[SPACE]dataregels![SPAC
     E](vanaf[SPACE]regel[SPACE]240)":b
     =0:end
90   print"data[SPACE]is[SPACE]weggezet"
95   print"checksum[SPACE]testen[SPACE]
     met[SPACE]sys49152"
100  data 165,43,166,44,133,163,134,164
     ,169,147
110  data 32,210,255,160,0,240,3,32,73,192
120  data 32,73,192,208,1,96,32,225,255,208
130  data 3,76,116,164,32,81,192,32,73,192
140  data 240,12,201,32,240,247,24,101,
     167,133
150  data 167,76,37,192,166,167,169,0,1
     32,168
160  data 32,205,189,169,13,32,210,255,
     164,168
170  data 76,17,192,200,208,2,230,164,1
     77,163
180  data 96,162,0,189,123,192,240,6,32,210
190  data 255,232,208,245,32,73,192,170
     ,32,73
200  data 192,132,168,32,205,189,162,3,
     169,32
210  data 32,210,255,202,208,250,169,0,
     133,167
220  data 164,168,96,82,69,71,69,76,32,0
230  data  -1
240  data 165,197,201,3,240,7,201,4,240
250  data 6,76,148,192,76,34,192,169
260  data 147,32,210,255,76,161,192
270  data -1
```

** EINDE LISTING checksum 64 **

chechsum checksum 64

REGEL 1	249	REGEL 110	158
REGEL 2	84	REGEL 120	232
REGEL 3	105	REGEL 130	183
REGEL 4	2	REGEL 140	96
REGEL 5	246	REGEL 150	96
REGEL 6	152	REGEL 160	127
REGEL 7	249	REGEL 170	71
REGEL 10	157	REGEL 180	223
REGEL 20	64	REGEL 190	73
REGEL 30	38	REGEL 200	79
REGEL 40	57	REGEL 210	109
REGEL 50	14	REGEL 220	106
REGEL 55	251	REGEL 230	225
REGEL 60	192	REGEL 240	16
REGEL 70	42	REGEL 250	163
REGEL 80	244	REGEL 260	92
REGEL 90	245	REGEL 270	22
REGEL 95	237		
REGEL 100	183		

PRINT OUT C-64 met o.a. Yathzee

SpriteHunter

De eigenlijke naam van dit programma is Sprite Hunter en deze naam verteld al veel over de werking. Het is met dit programma mogelijk sprites "uit" een programma te lichten. Het programma staat op adres $4001 weggeschreven, omdat een groot aantal programma's sprites in het basicgeheugen ($0801 - $1000) zetten. Als er geen cartridge is gebruikt kan ook vanaf adres $0801 worden geladen. Wanneer het moet gebeuren van adres $04001 typt U in voor het laden:
POKE 642,64:POKE44,64:poke16384,0:NEW
Verdere uitleg wordt er bij het opstarten van het programma op het beeldscherm weergegeven.

```
10  rem voor het intikken of laden van
20  rem dit programma tik je:
30  rem poke642,64:poke44,64:poke16384
    ,0:          new
40  rem anders zullen enkele sprites
50  rem misschien gewist worden !
60  :
70  rem ********************************
80  rem ****     sprite hunter      ****
90  rem ****    1989 by pitsoft     ****
100 rem********************************
110 rem
120 poke53281,0:poke53280,0:print"[SHI
    FT CLR][CTRL-5][CTRL-N]";:poke5326
    9,0:goto590
130 v=53248:pokev+21,1:pokev+28,0:poke
    v+23,1:pokev+29,1:pokev,100:pokev+
    1,150
140 bs=32:poke2040,bs:pokev+37,2:pokev
    +38,5:pokev+16,0:dimpo%(513):poke6
    50,128
150 print"[SHIFT-CLR]"chr$(8)"[CTRL-9]
    OPTIONS:[7xSPACE]"
160 print"[CTRL-9][SPACE]1)[SPACE]Mult
    icolor[SPACE]"
170 print"[CTRL-9][SPACE]2)[SPACE]Sing
    lecolor"
180 print"[CTRL-9][SPACE]3)[SPACE]Forw
    ard[4xSPACE]"
190 print"[CTRL-9][SPACE]4)[SPACE]Back
    ward[3xSPACE]"
200 print"[CTRL-9][SPACE]5)[SPACE]Jump
    [7xSPACE]"
210 print"[CTRL-9][SPACE]6)[SPACE]Colo
    r[6xSPACE]"
220 print"[CTRL-9][SPACE]7)[SPACE]Sele
    ct[5xSPACE]"
230 print"[CTRL-9][SPACE]8)[SPACE]Dese
    lect[3xSPACE]"
240 print"[CTRL-9][SPACE]9)[SPACE]Save
    [7xSPACE]"
250 print"[HOME]"tab(29)"sn[5xSPACE][4
    xCRSR-LEFT]"bs:geta$
260 poke214,15:print:printpo%(bs):a=va
    l(a$):ifa<1ora>9then250
270 onagoto280,290,300,320,340,380,410
    ,420,430
280 pokev+28,1:goto250
290 pokev+28,0:goto250
300 ifbs<512thenbs=bs+1:poke2040,bsand
    255:ifbs>255thengosub570
310 goto250
320 ifbs>=33thenbs=bs-1:poke2040,bsand
    255:ifbs>255thengosub570
330 goto250
340 poke214,17:print:input"Sprite-numm
    er";bs:ifbs<32orbs>512then340
350 poke214,17:print:print"[SHIFT-SPAC
    E][21xSPACE][6xSHIFT-SPACE]"
360 poke2040,bsand255:ifbs>255thengosu
    b570
370 goto150
380 poke214,17:print:input"col1";c1:in
    put"col2";c2:input"col3";c3
390 ifc1<0orc1>255orc2<0orc2>255orc3<0
    orc3>255then380
400 pokev+39,c1:pokev+37,c2:pokev+38,c
    3:goto150
410 po%(bs)=1:goto250
420 po%(bs)=0:goto250
430 rem
440 rem  save gedeelte
450 rem
460 print"[SHIFT-CLR]Alle[SPACE]gesele
    cteerde[SPACE]sprites[SPACE]worden
    [SPACE]nu"
470 print"gesaved.Na[SPACE]het[SPACE]s
    aven[SPACE]kunt[SPACE]u[SPACE]de[S
    PACE]data"
480 print"laden[SPACE]met:load"chr$(34
    )"filenaam"chr$(34)",8,1"
490 print"De[SPACE]sprites[SPACE]bevin
    den[SPACE]zich[SPACE]dan[SPACE]van
    af[SPACE]$2000";
500 print"oftewel[SPACE]#8192."
510 print"FILENAAM[SPACE](return=quit)
    ";:inputf$:iff$=""thenrun
520 l=0:open1,8,2,"`:"+f$+",p,w":print
    #1,chr$(0);chr$(32);:n=32
530 ifpo%(n)=0then550
540 l=l+1:ifl<=127thenba=n*64:fork=0to
    63:print#1,chr$(peek(ba+k));:nextk
550 n=n+1:ifn<512then530
560 close1:run
570 rem ***sprite geheugen boven 255**
    *
580 ba=bs*64:poke2040,13:fork=0to63:po
    ke832+k,peek(ba+k):next:return
590 print"Welkom[SPACE]bij[SPACE]de[SP
    ACE]Sprite[SPACE]Hunter.Met[SPACE]
    dit"
600 print"programma[SPACE]kunt[SPACE]u
    [SPACE]de[SPACE]sprites[SPACE]uit[
    SPACE]het"
610 print"geheugen[SPACE]halen[SPACE]e
    n[SPACE]de[SPACE]gewenste[SPACE]sp
    rites"
630 print"saven.Linksboven[SPACE]het[S
    PACE]scherm[SPACE]staan[SPACE]de"
640 print"opties[SPACE]en[SPACE]rechts
    boven[SPACE]het[SPACE]sprite-nr."
650 print"Naast[SPACE]de[SPACE]actuele
    [SPACE]sprite[SPACE]staat[SPACE]ee
    n[SPACE]0[SPACE]of"
660 print"een[SPACE]1.Als[SPACE]er[SPA
    CE]een[SPACE]1[SPACE]staat[SPACE]i
    s[SPACE]de[SPACE]sprite"
670 print"geselecteerd[SPACE]voor[SPAC
    E]het[SPACE]SAVE'n."
```

```
680 print"U[SPACE]kunt[SPACE]sprites[S
    PACE]pakken[SPACE]die[SPACE]hoger[
    SPACE]liggen"
690 print"dan[SPACE]spritenummer[SPACE
    ]255,waar[SPACE]vaak[SPACE]sprites"
700 print"'verborgen[SPACE]liggen'[SPA
    CE]in[SPACE]Games."
710 print"U[SPACE]kunt[SPACE]de[SPACE]
    Multicolor[SPACE]mode[SPACE]ook[SP
    ACE]gebruiken.";
720 print"Het[SPACE]programma[SPACE]wo
    rdt[SPACE]vanaf[SPACE]$4001[SPACE]
    geladen"
730 print"in[SPACE]plaats[SPACE]van[SP
    ACE]$0801[SPACE]omdat[SPACE]anders
    [SPACE]mis-"
740 print"schien[SPACE]sprites[SPACE]w
    orden[SPACE]gewist.U[SPACE]kunt[SP
    ACE]het"
750 print"programma[SPACE]echter[SPACE
    ]ook[SPACE]vanaf[SPACE]$0801[SPACE
    ]laden."
760 print"[4xSPACE][CTRL-2]1989[SPACE]
    by[SPACE]wim[SPACE]de[SPACE]wild"
770 print"[4xSPACE][CTRL-3]press[SPACE
    ]any[SPACE]key[SPACE]!"
780 wait198,1:poke198,0:goto130
```

** EINDE LISTING hunter **

Checksum hunter

REGEL 10	241	REGEL 420	230
REGEL 20	199	REGEL 430	143
REGEL 30	70	REGEL 440	253
REGEL 40	4	REGEL 450	143
REGEL 50	245	REGEL 460	91
REGEL 60	58	REGEL 470	177
REGEL 70	207	REGEL 480	5
REGEL 80	140	REGEL 490	233
REGEL 90	126	REGEL 500	24
REGEL 100	207	REGEL 510	133
REGEL 110	143	REGEL 520	238
REGEL 120	75	REGEL 530	17
REGEL 130	10	REGEL 540	126
REGEL 140	111	REGEL 550	198
REGEL 150	252	REGEL 560	149
REGEL 160	211	REGEL 570	192
REGEL 170	11	REGEL 580	154
REGEL 180	224	REGEL 590	172
REGEL 190	11	REGEL 600	160
REGEL 200	9	REGEL 610	53
REGEL 210	77	REGEL 630	166
REGEL 220	143	REGEL 640	58
REGEL 230	25	REGEL 650	81
REGEL 240	0	REGEL 660	141
REGEL 250	85	REGEL 670	66
REGEL 260	187	REGEL 680	213
REGEL 270	12	REGEL 690	155
REGEL 280	184	REGEL 700	61
REGEL 290	183	REGEL 710	26
REGEL 300	227	REGEL 720	185
REGEL 310	32	REGEL 730	83
REGEL 320	98	REGEL 740	137
REGEL 330	32	REGEL 750	18
REGEL 340	254	REGEL 760	254
REGEL 350	12	REGEL 770	120
REGEL 360	224	REGEL 780	183
REGEL 370	31		
REGEL 380	222		
REGEL 390	139		
REGEL 400	182		
REGEL 410	231		

Mastermind

Ditmaal weer eens een overbekend spel in een computervariatie, Mastermind. De kleuren zijn bij deze uitvoering zelf te kiezen. De maker van deze hersen pijniger is Wim de Wild uit Nigtevecht.

```
10  rem ********************************
20  rem *       master mind / c64      *
30  rem *       ----------             *
40  rem *                              *
50  rem * programmed by wim de wild    *
60  rem *                              *
70  rem * a pitsoft production in 1990 *
80  rem ********************************
90  dima(16):printchr$(8)"[CTRL-N]":hi
    =1500:gosub1060
100 poke53280,9:poke53281,0:print"[SHI
    FT CLR][COM-2][CTRL-9][SPACE]Maste
    r[SPACE]Mind[SPACE]/[SPACE]PITSOFT
    [SPACE]1990[13xSPACE]"
110 poke214,3:print:fork=0to18:printta
    b(6)"[CTRL-9][4xCOM-M][COM-L][12xS
    PACE]":next
120 poke214,3:print:printtab(25)"[CTRL
     2]Kansen[SPACE]over[SPACE]:"
130 poke214,6:print:printtab(25)"Tijd:
    "
140 c=0:fory=0to9:forx=0to3:gosub790:n
    extx,y:y=y-1
150 poke214,1:print:input"Hoeveel[SPAC
    E]kleuren[SPACE]";a$:a=val(a$):ifa
    <4ora>14then150
160 a=val(a$):l=0:fork=1toa:ifk=9thenk
    =10:l=-1
170 n=1184+(k+l)*40:poken,160:poken+54
    272,k:a(k+l)=k:next
180 rem
190 rem kleuren bedenken
200 rem
210 fork=0to3
220 a1=int(rnd(0)*a)+1:forl=0tok:ifa1=
    cl(l)then220
230 nextl:cl(k)=a1:nextk:gosub970
240 poke214,1:print:print"[CTRL-2][3xS
    PACE]S[SPACE]p[SPACE]e[SPACE]l[2xS
    PACE]i[SPACE]s[2xSPACE]b[SPACE]e[S
    PACE]g[SPACE]o[SPACE]n[SPACE]n[SPA
    CE]e[SPACE]n[SPACE]!":ti$="000000"
250 rem
260 rem hoofdprogramma
270 rem
280 dx=1225:x=0
290 gosub860:pokedx,31:pokedx+54272,1
300 geta$:ifa$=""thengosub900:goto300
310 ifa$="b"then100
320 ifa$="h"then1060
330 ifa$<>"[CRSR-DOWN]"then350
340 ifdx<1185+40*(a+(a>9))thenpokedx,3
    2:dx=dx+40:goto290
350 ifa$<>"[CRSR-UP]"then370
360 ifdx>1225thenpokedx,32:dx=dx-40:go
    to290
370 ifa$<>chr$(13)then290
380 xl(x)=peek(dx+54271)and15:c=xl(x):
    gosub790
390 x=x+1:ifx=4thenx=0:pokedx,32:goto4
```

```
    10
400 goto290
410 fork=0to3:forl=0to3:ifx1(k)=x1(l)a
    ndk<>lthengosub940:goto280
420 nextl,k
430 x2=int(rnd(0)*4):fork=0to3:sp(k)=9
    :ifx1(k)=cl(k)thensp(k)=0:goto460
440 forl=0to3:ifx1(k)=cl(l)andl<>kthen
    sp(k)=1:goto460
450 nextl
460 nextk:b=55296+(y*2+4)*40+6:k=0
470 pokeb+x2,sp(k):k=k+1:x2=x2+1
480 ifx2=4thenx2=0
490 ifk<4then470
500 t=0:fork=0to3:t=t+sp(k):next:ift=0
    then600
510 y=y-1:ify>=0then280
520 print"[SHIFT-CLR][CTRL-N]Je[SPACE]
    hebt[SPACE]verloren[SPACE](vind[SP
    ACE]ik[SPACE]niet[SPACE]erg)"
530 a$(0)="Probeer[SPACE]eens[SPACE]ee
    n[SPACE]lager[SPACE]level,leeghoof
    d!"
540 a$(1)="Alleen[SPACE]apen[SPACE]van
    [SPACE]het[SPACE]lage[SPACE]soort[
    SPACE]presterendit."
550 a$(2)="Ik[SPACE]heb[SPACE]nog[SPAC
    E]nooit[SPACE]zo'n[SPACE]nutcase[S
    PACE]als[SPACE]jij[3xSPACE]gezien.
    (En[SPACE]zo[SPACE]zijn[SPACE]er[S
    PACE]meer)"
560 a$(3)="Toch[SPACE]wat[SPACE]moeili
    jker[SPACE]dan[SPACE]je[SPACE]dach
    t[SPACE]?[6xSPACE]YOU[SPACE]SLIME[
    SPACE]FREAK!"
570 a$(4)="Go[SPACE]back[SPACE]to[SPAC
    E]the[SPACE]junkyard[SPACE]where[S
    PACE]you[SPACE]belong"
580 a$(5)="Ga[SPACE]maar[SPACE]weer[SP
    ACE]terug[SPACE]naar[SPACE]Pacman[
    SPACE]om[SPACE]je[4xSPACE]schuld[S
    PACE]goed[SPACE]te[SPACE]maken."
590 m=int(rnd(0)*6):printa$(m):goto710
600 rem
610 rem hall of fame
620 rem
630 poke53281,1:poke53280,1:print"[SHI
    FT CLR][CTRL-3]"tab(10)"GEFELICITE
    ERD!"
640 printtab(10)"————————":sc=int(
    a*1000/val(ti$))*(a*2)
650 input"Hoe[SPACE]is[SPACE]uw[SPACE]
    naam";a$
660 printa$;",[SPACE]uw[SPACE]score[SP
    ACE]was"sc"!"
670 ifsc>hithenhi=sc:print"U[SPACE]ove
    rtrof[SPACE]daarmee[SPACE]de[SPACE
    ]hoogste[SPACE]score."
680 print"[3xCRSR-DOWN][CTRL-4][2xSPAC
    E](c)[SPACE]1990[SPACE]PITSOFT[SPA
    CE]Nigtevecht"
690 print"Dit[SPACE]programma[SPACE]is
    [SPACE]gemaakt[SPACE]door[SPACE]W.
    de[SPACE]Wild.";
700 print"26-2-'90[SPACE]voor[SPACE]Co
    mmodore[SPACE]Info."
710 wait198,1:poke198,0:goto100
720 rem
730 rem subroutine voor pion positie
740 rem y=kans , x=pion 0-3
750 b=55296+(y*2+4)*40+13+x*2:return
760 rem
770 rem subroutine voor pion plaatsen
780 rem c=kleur
790 gosub750:pokeb,c:return
800 rem subroutine voor kleur lezen
810 rem c wordt kleur
820 gosub750:c=peek(b)and15:return
830 rem
840 rem kansen
850 rem
860 poke214,4:print:printtab(27)y;"[CR
    SR-LEFT][2xSPACE]"
870 rem
880 rem tijd
890 rem
900 poke214,7:print:printtab(27)ti$:re
    turn
910 rem
920 rem error
930 rem
940 poke214,1:print:print"foute[SPACE]
    kleuren[SPACE]invoer[SPACE].[SPACE
    ]doe[SPACE]het[SPACE]over[SPACE]![
    2xSPACE]"
950 poke198,0:wait198,1:poke198,0:poke
    214,1:print
960 print"[39xSPACE]":return
970 rem
980 rem niveau
990 rem
1000 ifa<15thennv$="++"
1010 ifa<12thennv$="+"
1020 ifa<10thennv$="+-"
1030 ifa<8thennv$="-"
1040 ifa<5thennv$="--"
1050 poke214,11:print:printtab(25)"Nive
     au[SPACE]:":printtab(25)nv$:return
1060 rem
1070 rem  speluitleg
1080 rem
1090 poke53281,0:poke53280,7:print"[CTR
     L 8][SHIFT-CLR]";
1100 print"[2xSPACE]MASTER[SPACE]MIND[S
     PACE]by[SPACE]Wim[SPACE]de[SPACE]W
     ild"
1110 print"[2xSPACE]——————————-
     ——-"
1120 print"[CRSR-DOWN][SPACE]De[SPACE]b
     edoeling[SPACE]van[SPACE]dit[SPACE
     ]spel[SPACE]is[SPACE]om[SPACE]een"
1130 print"[SPACE]bepaalde[SPACE]kleurc
     ode[SPACE]te[SPACE]kraken.Je[SPACE
     ]krijgt"
1140 print"[SPACE]hiervoor[SPACE]9[SPAC
     E]kansen.Je[SPACE]kan[SPACE]mbv.[S
     PACE]de[SPACE][CTRL-9]crsr"
1150 print"[SPACE]toetsen[SPACE]en[SPAC
     E][CTRL-9]RETURN[CTRL-0][SPACE]een
     [SPACE]kleur[SPACE]selec-"
1160 print"[SPACE]teren.De[SPACE]code[S
     PACE]bestaat[SPACE]uit[SPACE]4[SPA
     CE]verschil-"
1170 print"[SPACE]lende[SPACE]kleuren[S
     PACE](je[SPACE]kunt[SPACE]kiezen[S
     PACE]tussen"
```

```
1180 print"[SPACE]4[SPACE]en[SPACE]14[S
     PACE]kleuren).Na[SPACE]4[SPACE]kle
     uren[SPACE]te[SPACE]heb-"
1190 print"[SPACE]ben[SPACE]geselecteer
     d[SPACE]geeft[SPACE]de[SPACE]compu
     ter[SPACE]een"
1200 print"[SPACE]aanwijzingen[SPACE]in
     [SPACE]de[SPACE]vorm[SPACE]van[SPA
     CE]pionnen."
1210 print"[CTRL-6][SPACE]Als[SPACE]je[
     SPACE]een[SPACE]zwarte[SPACE]pion[
     SPACE]krijgt[SPACE]dan[SPACE]staat
     ";
1220 print"[SPACE]er[SPACE]een[SPACE]kl
     eur[SPACE]op[SPACE]de[SPACE]goede[
     SPACE]plaats."
1230 print"[SPACE]Als[SPACE]je[SPACE]ee
     n[SPACE]witte[SPACE]pion[SPACE]kri
     jgt[SPACE]heb[SPACE]je[SPACE]een";
1240 print"[SPACE]kleur[SPACE]die[SPACE
     ]in[SPACE]de[SPACE]code[SPACE]voor
     komt[SPACE]maar[SPACE]je"
1250 print"[SHIFT-SPACE]hebt[SPACE]hem[
     SPACE]dan[SPACE]niet[SPACE]op[SPAC
     E]de[SPACE]goede[SPACE]plaats."
1260 print"[CTRL-5][SPACE]Als[SPACE]je[
     SPACE]4[SPACE]zwarte[SPACE]pionnen
     [SPACE]hebt[SPACE]dan[SPACE]heb"
1270 print"[SPACE]je[SPACE]gewonnen."
1280 print"[CTRL-7]NB.[SPACE]De[SPACE]p
     ionnen[SPACE]staan[SPACE]niet[SPAC
     E]op[SPACE]de[SPACE]overeen-";
1290 print"komstige[SPACE]plaats[SPACE]
     van[SPACE]het[SPACE]speelbord[SPAC
     E]!!!"
1300 goto710
```

EINDE LISTING master mind

Checksum master mind

REGEL 10	207	REGEL 770	115
REGEL 20	179	REGEL 780	146
REGEL 30	224	REGEL 790	115
REGEL 40	227	REGEL 800	230
REGEL 50	18	REGEL 810	229
REGEL 60	227	REGEL 820	138
REGEL 70	190	REGEL 830	143
REGEL 80	207	REGEL 840	79
REGEL 90	197	REGEL 850	143
REGEL 100	30	REGEL 860	222
REGEL 110	142	REGEL 870	143
REGEL 120	101	REGEL 880	186
REGEL 130	146	REGEL 890	143
REGEL 140	62	REGEL 900	245
REGEL 150	120	REGEL 910	143
REGEL 160	184	REGEL 920	25
REGEL 170	6	REGEL 930	143
REGEL 180	143	REGEL 940	37
REGEL 190	225	REGEL 950	199
REGEL 200	143	REGEL 960	165
REGEL 210	133	REGEL 970	143
REGEL 220	86	REGEL 980	87
REGEL 230	197	REGEL 990	143
REGEL 240	72	REGEL 1000	160
REGEL 250	143	REGEL 1010	114
REGEL 260	165	REGEL 1020	157
REGEL 270	143	REGEL 1030	73
REGEL 280	140	REGEL 1040	115
REGEL 290	160	REGEL 1050	241
REGEL 300	73	REGEL 1060	143
REGEL 310	96	REGEL 1070	141
REGEL 320	156	REGEL 1080	143
REGEL 330	232	REGEL 1090	175
REGEL 340	22	REGEL 1100	18
REGEL 350	106	REGEL 1110	111
REGEL 360	158	REGEL 1120	138
REGEL 370	18	REGEL 1130	6
REGEL 380	29	REGEL 1140	67
REGEL 390	119	REGEL 1150	10
REGEL 400	36	REGEL 1160	100
REGEL 410	223	REGEL 1170	252
REGEL 420	69	REGEL 1180	100
REGEL 430	109	REGEL 1190	89
REGEL 440	30	REGEL 1200	240
REGEL 450	206	REGEL 1210	43
REGEL 460	109	REGEL 1220	78
REGEL 470	176	REGEL 1230	163
REGEL 480	14	REGEL 1240	158
REGEL 490	255	REGEL 1250	252
REGEL 500	102	REGEL 1260	85
REGEL 510	50	REGEL 1270	251
REGEL 520	65	REGEL 1280	37
REGEL 530	43	REGEL 1290	206
REGEL 540	111	REGEL 1300	3
REGEL 550	20		
REGEL 560	234		
REGEL 570	34		
REGEL 580	86		
REGEL 590	84		
REGEL 600	143		
REGEL 610	94		
REGEL 620	143		
REGEL 630	171		
REGEL 640	252		
REGEL 650	42		
REGEL 660	183		
REGEL 670	129		
REGEL 680	48		
REGEL 690	85		
REGEL 700	166		
REGEL 710	180		
REGEL 720	143		
REGEL 730	56		
REGEL 740	217		
REGEL 750	213		
REGEL 760	143		

Multibase

Multibase is een database welke volledig menu-gestuurd is. Er wordt gewerkt met de pijltjes toetsen en de RETURN-toets. Het programma is zo duidelijk dat een verdere uitleg overbodig is. Het programma is bedacht en uitgewerkt door Wim de Wild uit Nigtevecht.

```
10   rem ********************************
     **
20   rem ***       multibase v1.0.1      *
     **
30   rem ***          pitsoft            *
     **
40   rem ********************************
     **
50   poke53281,0:poke53280,0:print"[SHI
     FT CLR]"tab(10)"[CTRL-N][SPACE][CT
     RL 8]MULTIBASE[SPACE]1.0":poke19,1
60   printtab(10)"[SPACE][CTRL-7][13xCO
     M T]"
70   print"[CTRL-3](c)[SPACE]juli[SPACE
     ]1990[SPACE]by[SPACE]Pitsoft."
80   print"[CTRL-4]Pitsoft[SPACE]is[SPA
     CE]Wim[SPACE]de[SPACE]Wild."
90   fork=0to2000:next
100  rem ***************************
110  rem *      menu afdeling      *
120  rem ***************************
130  print"[SHIFT-CLR]";
140  printtab(10)"[COM-8]MULTIBASE[SPAC
     E]1.0[CTRL-N]"+chr$(8)
150  printtab(10)"[COM-4][13xCOM-T][COM
      5]"
160  print"[2xCRSR-DOWN]MainMenu."
170  print"[CRSR-DOWN][2xSPACE]1)[SPACE
     ]Velden[SPACE]Initialiseren
180  print"[CRSR-DOWN][2xSPACE]2)[SPACE
     ]Printer[SPACE]Lay-Out
190  print"[CRSR-DOWN][2xSPACE]3)[SPACE
     ]Bestand[SPACE]aanleggen
200  print"[CRSR-DOWN][2xSPACE]4)[SPACE
     ]Disk[SPACE]menu
210  print"[CRSR-DOWN][2xSPACE]5)[SHIFT
      SPACE]Uitvoer
220  print"[CRSR-DOWN][2xSPACE]6)[SPACE
     ]Sorteren
230  hy=5:gosub2230:ony%goto270,420,610
     ,830,1540,2060
240  rem ***************************
250  rem *   veld initialisatie    *
260  rem ***************************
270  clr:print"[SHIFT-CLR]"tab(10)"[COM
      8]Velden[SPACE]Init."
280  printtab(10)"[COM-4][12xCOM-T]"
290  input"[CTRL-4][SPACE]Aantal[SPACE]
     velden[SPACE](max.15)[SPACE]";ve%:
     ifve%<1orve%>15then290
300  print"[CTRL-3][SPACE]Benodigde[SPA
     CE]bestanden[SPACE](max.";int(1000
     /ve%);")";:inputbe%
310  ifbe%<1orbe%>1000/ve%then300
320  dimbe$(be%+1,ve%+1),ve$(ve%)
330  fork=0tove%-1:print"veld[SPACE]"k+
     1;:inputve$(k):ifve$(k)=""thenve$(
     k)=chr$(1)
340  m=len(ve$(k)):ifm>15thenk=k-1
350  ifm>hlthenhl=m
360  next:hl=hl-1
370  forq=0tove%-1:m=hl-len(ve$(q)):for
     z=1tom:ve$(q)=ve$(q)+"[SPACE]":nex
     tz,q
380  init=1:goto130
390  rem ****************************
400  rem *       printer layout     *
410  rem ****************************
420  print"[SHIFT-CLR]"tab(10)"[COM-8]P
     rinter[SPACE]Layout"
430  printtab(10)"[COM-4][14xCOM-T]"
440  print"[SPACE]Als[SPACE]u[SPACE]een
     [SPACE]MPS-801[SPACE]of[SPACE]803[
     SPACE]heeft,hoeft"
450  print"[SPACE]u[SPACE]alleen[SPACE]
     maar[SPACE]op[SPACE]RETURN[SPACE]t
     e[SPACE]drukken."
460  input"Kolommen[4xSPACE]80[4xCRSR-L
     EFT]";ko%
470  ifko%<40orko%>200then460
480  input"Kleine-letter-code[4xSPACE]1
     7[4xCRSR-LEFT]";km%
490  ifkm%<0orkm%>255then480
500  input"Bladlengte[4xSPACE]72[4xCRSR
     -LEFT]";bl%:ifbl%<10then500
510  input"Dubbel-breed[SPACE]print[4xS
     PACE]14[4xCRSR-LEFT]";db%:ifdb%<0o
     rdb%>255then510
520  input"Normaal-print[4xSPACE]15[4xC
     RSR-LEFT]";np%:ifnp%<0ornp%>255the
     n520
530  input"Tabulator[SPACE](beginpositi
     e)[4xSPACE]0[4xCRSR-LEFT]";ta%:ift
     a%<0orta%>ko%-1then530
540  input"PrintDevice[4xSPACE]4[4xCRSR
     -LEFT]";pd%
550  print"[2xCRSR-DOWN][CTRL-2]Voor[SP
     ACE]een[SPACE]Lay-Out[SPACE]librar
     y[SPACE]naar[SPACE]Disk[SPACE]menu
     ."
560  fork=0to1000:next
570  goto130
580  rem ****************************
590  rem *     invoer afdeling      *
600  rem ****************************
610  ifinit=0then2340
620  print"[SHIFT-CLR]"tab(10)"[COM-8]B
     ESTAND[SPACE]AANLEGGEN"
630  printtab(10)"[COM-4][17xCOM-T]"
640  print"[HOME][CTRL-3]N"nr%"[CRSR-DO
     WN][COM-5]"
650  fork=0tove%-1:printve$(k)":":next
660  print"[HOME][CRSR-DOWN]"
670  z=0:forq=0tove%-1
680  printtab(len(ve$(q)));:inputbe$(nr
     %,q):be$(nr%,q)=be$(nr%,q)+chr$(1)
690  z=z-1:iflen(be$(nr%,q))+len(ve$(q)
     )+3>40thenz=1
700  nextq
710  ifz<>-qthen620
720  print"[CTRL-2][7xSPACE][CTRL-9]Zo[
     SPACE]goed[SPACE](j/n)[CTRL-0]
730  geta$:ifa$<>"j"anda$<>"n"then730
740  ifa$="n"then620
750  print"[SHIFT-SPACE][5xSPACE][CTRL
     9]Meer[SPACE]invoeren[SPACE](j/n[C
     TRL 0]
```

```
760  geta$:ifa$<>"j"anda$<>"n"then760
770  ifa$="n"thennr%=nr%+1:goto 130
780  nr%=nr%+1:ifnr%<=be%then620
790  print"[SHIFT-CLR][COM-8][2xCRSR-DO
     WN][7xSPACE]Hele[SPACE]bestand[SPA
     CE]is[SPACE]VOL!":goto2350
800  rem *************************
810  rem *     disk afdeling      *
820  rem *************************
830  print"[SHIFT-CLR]"tab(10)"[COM-8]M
     ULTIBASE[SPACE]1.0"
840  printtab(10)"[COM-4][13xCOM-T]"
850  print"[2xCRSR-DOWN]SubMenu[SPACE]1
     [SPACE]:[SPACE][COM-5]DiskMenu[COM
      4]
860  print"[CRSR-DOWN][SPACE]1)[SPACE]S
     aven[SPACE]Bestand
870  print"[CRSR-DOWN][SPACE]2)[SPACE]L
     aden[SPACE]Bestand
880  print"[CRSR-DOWN][SPACE]3)[SPACE]S
     aven[SPACE]Layout
890  print"[CRSR-DOWN][SPACE]4)[SPACE]L
     aden[SPACE]Layout
900  print"[CRSR-DOWN][SPACE]5)[SPACE]M
     ain[SPACE]Menu
910  x=0:hy=4:gosub2230
920  ony%goto960,1130,1300,1420,130
930  rem ***************************
940  rem *    bestanden saven       *
950  rem ***************************
960  print"[SHIFT-CLR][CTRL-3]"nr%;"[HO
     ME]"tab(10)"[COM-8]SavenBestand"
970  printtab(10)"[COM-4][12xCOM-T]"
980  if init=0then2340
990  input"Filenaam";f$
1000 iflen(f$)=0then990
1010 f$="`0:"+f$+",s,w"
1020 open1,8,2,f$
1030 print#1,"app[SPACE]data"
1040 print#1,ve%:print#1,nr%:print#1,be
     %
1050 fork=0tove%-1:print#1,ve$(k):next
1060 fork=0tonr%:forn=0tove%-1
1070 print#1,be$(k,n):next
1080 print"[HOME][6xCRSR-DOWN]"k:next
1090 close1:print"File[SPACE]is[SPACE]g
     esaved.":goto2350
1100 rem ***************************
1110 rem *    bestanden laden       *
1120 rem ***************************
1130 print"[SHIFT-CLR]"tab(10)"[COM-8]L
     aden[SPACE]Bestand"
1140 printtab(10)"[COM-4][13xCOM-T]"
1150 clr:input"Filenaam";f$
1160 iflen(f$)=0then1150
1170 f$=f$+",s,r"
1180 open5,8,15:open1,8,2,f$
1190 input#5,a,a$:ifa$<>"ok"thenclose5:
     close1:goto130
1200 input#1,a$:ifa$="app[SPACE]data"th
     en1220
1210 print"[SHIFT-CLR]Wrong[SPACE]FILE[
     SPACE]type.":close5:close1:goto235
     0
1220 input#1,ve%:input#1,nr%:input#1,be
     %:print"[HOME][CTRL-3]"nr%-1:dimbe
     $(be%,ve%),ve$(ve%)
1230 fork=0tove%-1:input#1,ve$(k):next
1240 fork=0tonr%-1:forn=0tove%-1
1250 input#1,be$(k,n):next
1260 print"[HOME][7xCRSR-DOWN]"k:next:c
     lose5:close1:in=1:nr%=nr%+1:goto13
     0
1270 rem *****************************
1280 rem *       saven layout         *
1290 rem *****************************
1300 print"[SHIFT-CLR]"tab(10)"[COM-8]S
     aven[SPACE]Layout"
1310 printtab(10)"[COM-4][12xCOM-T]"
1320 input"Filenaam[SPACE]";f$
1330 iflen(f$)=0then1320
1340 f$="`0:"+f$+",s,w"
1350 open1,8,2,f$
1360 print#1,"layout[SPACE]data"
1370 print#1,ko%:print#1,km%:print#1,db
     %:print#1,np%:print#1,ta%:print#1,
     bl%
1380 print#1,pd%:close1:goto130
1390 rem *****************************
1400 rem *      laden layout          *
1410 rem *****************************
1420 print"[SHIFT-CLR]"tab(10)"[COM-8]L
     aden[SPACE]Layout"
1430 printtab(10)"[COM-4][12xCOM-T]"
1440 input"Filenaam[SPACE]";f$
1450 iflen(f$)=0then1320
1460 f$=f$+",s,r"
1470 open1,8,2,f$
1480 input#1,a$:ifa$<>"layout[SPACE]dat
     a"then1210
1490 input#1,ko%:input#1,km%:input#1,db
     %:input#1,np%:input#1,ta%:input#1,
     bl%
1500 input#1,pd%:close1:goto130
1510 rem *****************************
1520 rem *         uitvoeren          *
1530 rem *****************************
1540 print"[SHIFT-CLR]"tab(10)"[COM-8]U
     itvoer"
1550 ifinit=0then2340
1560 printtab(10)"[COM-4][7xCOM-T]"
1570 print"[2xCRSR-DOWN][COM-4]SubMenu[
     SPACE]2[SPACE]:Uitvoer
1580 print"[CRSR-DOWN][SPACE]1)[SPACE]B
     eeldscherm
1590 print"[CRSR-DOWN][SPACE]2)[SPACE]P
     rinter
1600 x=0:hy=1:gosub2230
1610 ony%goto1620,1900
1620 print"[SHIFT-CLR][4xCRSR-DOWN]SubM
     enu[SPACE]3[SPACE]:Beeldscherm
1630 print"[CRSR-DOWN][SPACE]1)[SPACE]R
     ecord[SPACE]zoeken
1640 print"[CRSR-DOWN][SPACE]2)[SPACE]H
     ele[SPACE]bestand[SPACE]doorlopen
1650 x=0:hy=1:de=3:gosub2230
1660 ony%goto1670,1790
1670 print"[SHIFT-CLR][5xCRSR-DOWN][COM
      4]";
1680 fork=0tove%-1:print"[CRSR-DOWN]"k+
     1;ve$(k):next
1690 hy=k-1:x=0:gosub2230
1700 print"[SHIFT-CLR][COM-4]Welke[SPAC
     E]";ve$(y%-1);:inputn$:k%=0:n$=n$+
     chr$(1)
1710 ifbe$(k%,y%-1)=n$then1740
```

```
1720 k%=k%+1:ifk%<nr%then1710
1730 goto130
1740 print"[SHIFT-CLR]":forn=0tove%:pri
     ntve$(n);be$(k%,n):next
1750 print"[CRSR-DOWN][5xCRSR-RIGHT]ver
     der[SPACE]zoeken[SPACE]naar[SPACE]
     nog[SPACE]een"chr$(13)"[5xCRSR-RIG
     HT]"n$"?"
1760 geta$:ifa$<>"j"anda$<>"n"then1760
1770 ifa$="j"then1720
1780 goto130
1790 print"[SHIFT-CLR]";
1800 n%=0
1810 form=0tonr%-1:fork=0tove%-1:printv
     e$(k);"[SPACE]";be$(m,k)
1820 next:print"Bestand";m
1830 geta$:ifa$=""then1830
1840 ifa$="[PIJL LINKS]"then130
1850 print"[SHIFT-CLR]":nextm
1860 goto130
1870 rem ****************************
1880 rem *       printer afdeling     *
1890 rem ****************************
1900 print"[SHIFT-CLR]"tab(10)"Printer[
     SPACE]Uitvoer"
1910 ifko%=0thenprint"Geen[SPACE]Printe
     r[SPACE]Layout[SPACE]aanwezig.":go
     to2350
1920 open4,pd%:print#4:close4:ifst<>0th
     enprint"Geen[SPACE]printer[SPACE]a
     anwezig.":goto2350
1930 open4,pd%:print#4,chr$(km%)chr$(db
     %);:pa=1
1940 print#4,"MULTIBASE[SPACE]1.0.x[SPA
     CE]1990[SPACE]Wim[SPACE]de[SPACE]W
     ild":print#4,chr$(np%)chr$(km%);:l
     e=4
1950 h%=int(ko%/40):n%=40*h%+ta%:ifn%>k
     o%thenh%=h%-1
1960 k$=chr$(km%):forn=0tonr%steph%:for
     k=0tove%-1
1970 print#4,spc(ta%)k$;:fori=1toh%:q=4
     0-(len(ve$(k))+len(be$(n+(i-1),k))
     +1)
1980 print#4,ve$(k)"[SPACE]"be$(n+(i-1)
     ,k);spc(q);:next:print#4:gosub2380
     :next
1990 print#4:gosub2380:next
2000 fork=le tobl%:print#4:next:gosub23
     80
2010 close4
2020 print"[3xCRSR-DOWN][4xCRSR-RIGHT]B
     estand[SPACE]is[SPACE]VOLLEDIG[SPA
     CE]uitgeprint.":goto2350
2030 rem ****************************
2040 rem *    sorteer afdeling       *
2050 rem ****************************
2060 print"[SHIFT-CLR]"tab(10)"[COM-8]S
     orteren"
2070 printtab(10)"[COM-4][8xCOM-T]"
2080 if init=0then2340
2090 print"WAARSCHUWING:"
2100 print"Het[SPACE]sorteren[SPACE]bij
     [SPACE]grote[SPACE]bestanden[SPACE
     ]kost"
2110 print"letterlijk[SPACE]uren[SPACE]
     tijd.U[SPACE]kunt[SPACE]terug[SPAC
     E]naar"
2120 print"het[SPACE]MENU[SPACE]als[SPA
     CE]u[SPACE]op[SPACE][PIJL LINKS][S
      PACE]drukt."
2130 geta$:ifa$=""then2130
2140 ifa$="[PIJL LINKS]"then130
2150 print"[SHIFT-CLR][4xCRSR-DOWN]Kies
     [SPACE]veld."
2160 fork=0tove%-1:print"[CRSR-DOWN]"ve
     $(k):next
2170 x=0:hy=ve%-1:gosub2230:y%=y%-1:pri
     nty%
2180 forb=0tonr%-1:forc=btonr%-1:ifbe$(
     b,y%)>be$(c,y%)thengosub2430
2190 nextc,b:goto130
2200 rem ****************************
2210 rem * !!    menu afdeling    !! *
2220 rem ****************************
2230 x=30:y%=0
2240 y1%=5+y%*2
2250 poke214,y1%:print:poke211,x:print"
     [CTRL-2][PIJL LINKS]"
2260  geta$:ifa$<>"[CRSR-DOWN]"anda$<>"[
     CRSR-UP]"anda$<>chr$(13)then2260
2270 poke214,y1%:print:poke211,x:print"
     [SPACE]" :ifa$="[CRSR-UP]"theny%=y
     %-1:ify%<0theny%=0
2280 ifa$="[CRSR-DOWN]"theny%=y%+1:ify%
     >hytheny%=hy
2290 ifa$=chr$(13)theny%=y%+1:return
2300 goto2240
2310 rem ****************************
2320 rem *      geen velden !        *
2330 rem ****************************
2340 print"[SHIFT-CLR][2xSPACE][COM-8]N
     og[SPACE]geen[SPACE]velden[SPACE]g
     einitialiseerd[SPACE]!"
2350 print"[COM-5][9xSPACE][CTRL-9]Druk
     [SPACE]RETURN[CTRL-0]
2360 geta$:ifa$<>chr$(13)then2360
2370 goto130
2380 le=le+1:ifle<bl%thenreturn
2390 rem ****************************
2400 rem *       subroutines         *
2410 rem ****************************
2420 m$=str$(pa):l=len(m$):print#4,spc(
     ko%-1-2)m$:pa=pa+1:le=2:return
2430 fork=0tove%-1:aa$=be$(b,k):be$(b,k
     )=be$(c,k):be$(c,k)=aa$:next:retur
     n
2440 rem
2450 rem      restart-up
2460 rem
2470 ifin=1thengoto50
```

EINDE LISTING multibase

Checksum minibase

REGEL 10	207
REGEL 20	117
REGEL 30	180
REGEL 40	207
REGEL 50	57
REGEL 60	112
REGEL 70	6
REGEL 80	149
REGEL 90	208
REGEL 100	211
REGEL 110	82
REGEL 120	211
REGEL 130	171
REGEL 140	98
REGEL 150	128
REGEL 160	135
REGEL 170	180
REGEL 180	214
REGEL 190	43
REGEL 200	9
REGEL 210	120
REGEL 220	29
REGEL 230	183
REGEL 240	253
REGEL 250	215
REGEL 260	211
REGEL 270	114
REGEL 280	69
REGEL 290	79
REGEL 300	135
REGEL 310	144
REGEL 320	204
REGEL 330	190
REGEL 340	98
REGEL 350	87
REGEL 360	114
REGEL 370	249
REGEL 380	110
REGEL 390	39
REGEL 400	229
REGEL 410	39
REGEL 420	126
REGEL 430	139
REGEL 440	222
REGEL 450	94
REGEL 460	129
REGEL 470	84
REGEL 480	26
REGEL 490	40
REGEL 500	174
REGEL 510	10
REGEL 520	78
REGEL 530	186
REGEL 540	162
REGEL 550	242
REGEL 560	207
REGEL 570	29
REGEL 580	39
REGEL 590	240
REGEL 600	39
REGEL 610	17
REGEL 620	255
REGEL 630	116
REGEL 640	140
REGEL 650	86
REGEL 660	1
REGEL 670	106
REGEL 680	109
REGEL 690	78
REGEL 700	211
REGEL 710	132
REGEL 720	196
REGEL 730	109
REGEL 740	115
REGEL 750	253
REGEL 760	112
REGEL 770	73
REGEL 780	241
REGEL 790	135
REGEL 800	169
REGEL 810	72
REGEL 820	169
REGEL 830	49
REGEL 840	232
REGEL 850	246
REGEL 860	164
REGEL 870	140
REGEL 880	131
REGEL 890	107
REGEL 900	132
REGEL 910	137
REGEL 920	203
REGEL 930	253
REGEL 940	244
REGEL 950	253
REGEL 960	109
REGEL 970	69
REGEL 980	17
REGEL 990	43
REGEL 1000	52
REGEL 1010	14
REGEL 1020	40
REGEL 1030	52
REGEL 1040	132
REGEL 1050	52
REGEL 1060	66
REGEL 1070	114
REGEL 1080	93
REGEL 1090	222
REGEL 1100	253
REGEL 1110	219
REGEL 1120	253
REGEL 1130	225
REGEL 1140	232
REGEL 1150	1
REGEL 1160	89
REGEL 1170	113
REGEL 1180	44
REGEL 1190	1
REGEL 1200	205
REGEL 1210	180
REGEL 1220	127
REGEL 1230	32
REGEL 1240	30
REGEL 1250	94
REGEL 1260	228
REGEL 1270	81
REGEL 1280	62
REGEL 1290	81
REGEL 1300	215
REGEL 1310	69
REGEL 1320	43
REGEL 1330	88
REGEL 1340	14
REGEL 1350	40
REGEL 1360	49
REGEL 1370	55
REGEL 1380	16
REGEL 1390	81
REGEL 1400	37
REGEL 1410	81
REGEL 1420	190
REGEL 1430	69
REGEL 1440	43
REGEL 1450	88
REGEL 1460	113
REGEL 1470	40
REGEL 1480	123
REGEL 1490	191
REGEL 1500	252
REGEL 1510	81
REGEL 1520	164
REGEL 1530	81
REGEL 1540	42
REGEL 1550	17
REGEL 1560	22
REGEL 1570	173
REGEL 1580	196
REGEL 1590	203
REGEL 1600	134
REGEL 1610	87
REGEL 1620	188
REGEL 1630	49
REGEL 1640	120
REGEL 1650	46
REGEL 1660	99
REGEL 1670	151
REGEL 1680	142
REGEL 1690	124
REGEL 1700	95
REGEL 1710	20
REGEL 1720	138
REGEL 1730	29
REGEL 1740	205
REGEL 1750	55
REGEL 1760	161
REGEL 1770	161
REGEL 1780	29
REGEL 1790	171
REGEL 1800	85
REGEL 1810	197
REGEL 1820	162
REGEL 1830	153
REGEL 1840	128
REGEL 1850	121
REGEL 1860	29
REGEL 1870	81
REGEL 1880	65
REGEL 1890	81
REGEL 1900	51
REGEL 1910	98
REGEL 1920	36
REGEL 1930	107
REGEL 1940	190
REGEL 1950	236
REGEL 1960	136
REGEL 1970	169
REGEL 1980	32
REGEL 1990	28
REGEL 2000	188
REGEL 2010	212
REGEL 2020	189
REGEL 2030	81
REGEL 2040	65
REGEL 2050	81
REGEL 2060	110
REGEL 2070	185
REGEL 2080	17
REGEL 2090	170
REGEL 2100	219
REGEL 2110	42
REGEL 2120	222
REGEL 2130	147
REGEL 2140	128
REGEL 2150	185
REGEL 2160	45
REGEL 2170	6
REGEL 2180	101
REGEL 2190	138
REGEL 2200	81
REGEL 2210	214
REGEL 2220	81
REGEL 2230	7
REGEL 2240	156
REGEL 2250	64
REGEL 2260	155
REGEL 2270	235
REGEL 2280	52
REGEL 2290	22
REGEL 2300	81
REGEL 2310	81
REGEL 2320	225
REGEL 2330	81
REGEL 2340	191
REGEL 2350	141
REGEL 2360	130
REGEL 2370	29
REGEL 2380	160
REGEL 2390	81
REGEL 2400	70
REGEL 2410	81
REGEL 2420	235
REGEL 2430	88
REGEL 2440	143
REGEL 2450	134
REGEL 2460	143
REGEL 2470	15

Schermsplitsdemo

Het volgende programma is een demo. Het laadt zien wat er mogelijk is met teksten op een beeldscherm. De bedoeling hiervan is dat (toekomstige) programmeurs hier stukjes van kunnen gebruiken om speciale effekten te kreeren in hun eigen programma's. Schemsplitsdemo deelt het scherm in twee delen waarbij het onderste deel stukjes op en neer beweegt. De geschreven door Marco Paauwe uit Gorichem.

Schermsplitsdemo

```
10   poke53280,0:poke53281,0:poke646,7
20   printchr$(147);"[7xSPACE]gemaakt[S
     PACE]door[SPACE]marco[SPACE]paauwe
     "
30   print"[10xSPACE]dr.hiemstralaan[SP
     ACE]1[SPACE]"
40   print"[14xSPACE]4205[SPACE]kj"
50   print"[13xSPACE]gorinchem"
60   print"[2xCRSR-DOWN][2xSPACE]de[SPA
     CE]eerste[SPACE]6[SPACE]regels[SPA
     CE]blijven[SPACE]staan[SPACE]en"
70   print"[10xSPACE]scrollt[SPACE]vert
     icaal"
80   print"de[SPACE]tekst[SPACE]is[SPAC
     E]aan[SPACE]te[SPACE]passen[SPACE]
     door[SPACE]de[SPACE]print"
90   print"[8xSPACE]regels[SPACE]te[SPA
     CE]veranderen"
100  fori=49152to49331
110  reada:pokei,a:next
120  sys49152
130  print:print"met[SPACE]poke49237[SP
     ACE]kun[SPACE]je[SPACE]bepalen[SPA
     CE]hoever[SPACE]de[SPACE]"
140  print"[10xSPACE]tekst[SPACE]moet[S
     PACE]zakken"
150  gosub560:printchr$(13);"voorbeelde
     n"
160  poke49237,10:print"nu[SPACE]zakt[S
     PACE]het[SPACE]scherm[SPACE]niet[S
     PACE]ver(poke49237,10)":gosub560
170  poke49237,200:print"nu[SPACE]zakt[
     SPACE]het[SPACE]scherm[SPACE]heel[
     SPACE]wat[SPACE]verder(200)":gosub
     560
180  print"poke49237,145=normaal":poke4
     9237,145
190  goto190
200  data120,162,44,160,192
210  data142,20,3,140,21
220  data3,169,127,141,13
230  data220,169,1,141,26
240  data208,169,95,141,18
250  data208,173,17,208,41
260  data127,141,17,208,169
270  data0,133,2,169,0
280  data133,3,88,96,169
290  data1,141,25,208,165
300  data3,208,42,166,2
310  data240,24,160,0,173
320  data18,208,205,18,208
330  data240,251,41,7,9
340  data16,141,17,208,200
350  data132,2,228,2,176
360  data234,232,134,2,224
370  data145,208,4,169,1
380  data133,3,76,132,192
390  data166,2,160,0,173
400  data18,208,205,18,208
410  data240,251,41,7,9
420  data16,141,17,208,200
430  data132,2,228,2,176
440  data234,202,134,2,224
450  data1,176,4,169,0
460  data133,3,162,150,160
470  data192,142,20,3,140
480  data21,3,169,0,141
490  data18,208,76,49,234
500  data169,1,141,25,208
510  data169,23,141,17,208
520  data162,44,160,192,142
530  data20,3,140,21,3
540  data169,95,141,18,208
550  data76,129,234,192,165
560  fora=.to5000:nexta:return
```

** EINDE LISTING schermsplitsdemo demo **

checksum splitsdemo

REGEL 10	249	REGEL 400	208
REGEL 20	47	REGEL 410	54
REGEL 30	75	REGEL 420	196
REGEL 40	61	REGEL 430	103
REGEL 50	121	REGEL 440	194
REGEL 60	92	REGEL 450	6
REGEL 70	155	REGEL 460	195
REGEL 80	10	REGEL 470	144
REGEL 90	34	REGEL 480	47
REGEL 100	41	REGEL 490	169
REGEL 110	11	REGEL 500	155
REGEL 120	163	REGEL 510	208
REGEL 130	148	REGEL 520	254
REGEL 140	97	REGEL 530	243
REGEL 150	43	REGEL 540	218
REGEL 160	89	REGEL 550	13
REGEL 170	12	REGEL 560	20
REGEL 180	194		
REGEL 190	35		
REGEL 200	250		
REGEL 210	87		
REGEL 220	154		
REGEL 230	150		
REGEL 240	218		
REGEL 250	207		
REGEL 260	5		
REGEL 270	252		
REGEL 280	124		
REGEL 290	151		
REGEL 300	53		
REGEL 310	145		
REGEL 320	208		
REGEL 330	54		
REGEL 340	196		
REGEL 350	103		
REGEL 360	197		
REGEL 370	108		
REGEL 380	156		
REGEL 390	100		

Yathzee

Nu de avonden weer langzaam langer worden blijven de oude bord en kaartspelen de programmeurs trekken. Ditmaal heeft M. Boersma een computervariatie op het dobbelspel Yathzee gemaakt. Het spel heeft een aantal voordelen op het dobbelspel, er behoeft niet geteld te worden. Er is echter ook een nadeel, er kan niet worden valsgespeeld.

```
10  rem+++++++++++++++++++++++++++++
20  rem++                         ++
30  rem++      gemaakt door:      ++
40  rem++                         ++
50  rem++        m.boorsma         ++
60  rem++                         ++
70  rem++          voor           ++
80  rem++                         ++
90  rem++     commodore info      ++
100 rem+                          ++
110 rem+         (c) 1989         ++
120 rem+                          ++
130 rem++++++++++++++++++++++++++++++
200 s1=54272:s2=54279:s3=54286
210 fl=54293:fh=54294
220 rs=54295:pl=54296
230 pokes1+4,0:pokes2+4,0:pokes3+4,0
240 pokepl,15
250 pokes1+5,16*2+3 :pokes1+6,16*4+5
260 fori=1to100step5
270 pokes1,37+i:pokes1+1,17+i
280 pokes1+4,33:next
290 fori=100to1step-5
300 pokes1,37+i:pokes1+1,17+i
310 pokes1+4,17:next
320 pokes1+4,0:goto430
330 fori=100to1step-2.5
340 pokes1,37+i:pokes1+1,17+i
350 pokes1+4,17:next
360 pokes1+4,0:return
370 pokes1,50:pokes1+1,80
380 pokes1+4,33:fori=1to99:next
390 pokes1+4,0:return
400 pokes1,50:pokes1+1,50
410 pokes1+4,33:fori=1to99:next
420 pokes1+4,0:return
430 print"[SHIFT-CLR]":goto540
440 print"[HOME][2xCRSR-DOWN]";
450 print;a$(a);"[3xSPACE]Q[2xSPACE]Q[
    2xSPACE]QQ[2xSPACE]Q[2xSPACE]Q[SPA
    CE]QQQ[SPACE]QQQQ[SPACE]QQQQ[SPACE
    ]QQQQ
460 print"[3xSPACE]Q[2xSPACE]Q[SPACE]Q
    [2xSPACE]Q[SPACE]Q[2xSPACE]Q[2xSPA
    CE]Q[5xSPACE]Q[SPACE]Q[4xSPACE]Q
470 print"[4xSPACE]QQ[2xSPACE]QQQQ[SPA
    CE]QQQQ[2xSPACE]Q[3xSPACE]QQ[2xSPA
    CE]QQQ[2xSPACE]QQQ
480 print"[4xSPACE]QQ[2xSPACE]Q[2xSPAC
    E]Q[SPACE]Q[2xSPACE]Q[2xSPACE]Q[2x
    SPACE]Q[4xSPACE]Q[4xSPACE]Q
490 print"[4xSPACE]QQ[2xSPACE]Q[2xSPAC
    E]Q[SPACE]Q[2xSPACE]Q[2xSPACE]Q[2x
    SPACE]QQQQ[SPACE]QQQQ[SPACE]QQQQ
500 print:print:print
510 print;a$(b);"[9xSPACE]geprogrammee
    rd[SPACE]door[SPACE]:
520 print"[15xSPACE]m.boorsma
530 print"[17xSPACE]1989":return
540 a$(1)="[CTRL-8]":a$(2)="[COM-6]":a
    $(3)="[COM-1]":a=1
550 fori=1to15:ifa=1thena=3:goto570
560 ifa=3thena=1
570 b=int(rnd(1)*3+1)
580 gosub440
590 fory=1to250:next:next:b=0
600 print"[CTRL-8][8xCRSR-DOWN][10xSPA
    CE]start[SPACE]met[SPACE]toets[SPA
    CE]'s'
610 getq$:ifq$<>"s"then610
620 goto710
630 print"[SHIFT-CLR][COM-5][22xCRSR-D
    OWN][4xCRSR-RIGHT]er[SPACE]speelt[
    SPACE]nu[SPACE]:":print"[CTRL-4][6
    xCRSR-RIGHT]";sp$(n)
640 print"[HOME][CTRL-2][12xCRSR-RIGHT
    ]***[SPACE]yahtzee[SPACE]***":gosu
    b330
650 print"[HOME][CRSR-DOWN][CTRL-1][3x
    SPACE]U[3xSHIFT-*]I[2xSPACE]U[3xSH
    IFT *]I[2xSPACE]U[3xSHIFT-*]I[2xSP
    ACE]U[3xSHIFT-*]I[2xSPACE]U[3xSHIF
    T *]I
660 print"[3xSPACE][SHIFT -]Q[SPACE]Q[
    SHIFT -][2xSPACE][SHIFT -][3xSPACE
    ][SHIFT -][2xSPACE][SHIFT -]Q[2xSP
    ACE][SHIFT -][2xSPACE][SHIFT -]Q[S
    PACE]Q[SHIFT -][2xSPACE][SHIFT -]Q
    [SPACE]Q[SHIFT -]
670 print"[3xSPACE][SHIFT -]Q[SPACE]Q[
    SHIFT -][2xSPACE][SHIFT -][SPACE]Q
    [SPACE][SHIFT -][2xSPACE][SHIFT -]
    [SPACE]Q[SPACE][SHIFT -][2xSPACE][
    SHIFT -][3xSPACE][SHIFT -][2xSPACE
    ][SHIFT -][SPACE]Q[SPACE][SHIFT -]
680 print"[3xSPACE][SHIFT -]Q[SPACE]Q[
    SHIFT -][2xSPACE][SHIFT -][3xSPACE
    ][SHIFT -][2xSPACE][SHIFT -][2xSPA
    CE]Q[SHIFT -][2xSPACE][SHIFT -]Q[S
    PACE]Q[SHIFT -][2xSPACE][SHIFT -]Q
    [SPACE]Q[SHIFT -]
690 print"[3xSPACE]J[3xSHIFT-*]K[2xSPA
    CE]J[3xSHIFT-*]K[2xSPACE]J[3xSHIFT
     *]K[2xSPACE]J[3xSHIFT-*]K[2xSPACE
    ]J[3xSHIFT-*]K
700 print"[HOME][6xCRSR-DOWN][COM-8][5
    xSPACE]1[6xSPACE]2[6xSPACE]3[6xSPA
    CE]4[6xSPACE]5":goto790
710 print"[SHIFT-CLR][CTRL-8]eerst[SPA
    CE]het[SPACE]aantal[SPACE]spelers[
    SPACE]graag
720 input"[COM-1][CRSR-DOWN]aantal[SPA
    CE]spelers[SPACE](1-4)[SPACE]:[SPA
    CE]";s
730 ifs>4thens=4
740 k=s*13
750 print"[CTRL-8][CRSR-DOWN]graag[SPA
    CE]de[SPACE]namen[SPACE]van[SPACE]
    de[SPACE]spelers.[10xSPACE][CRSR-D
    OWN](max[SPACE]10[SPACE]letters)[C
    RSR-DOWN]
760 fori=1tos
770 print"[COM-6]speler";i;:inputsp$(i
    ):next
```

```
780  n=1:goto630
790  w$(1,1)="[3xSPACE]":w$(1,2)="[SPAC
     E]Q[SPACE]":w$(1,3)="[3xSPACE]"
800  w$(2,1)="Q[2xSPACE]":w$(2,2)="[3xS
     PACE]":w$(2,3)="[2xSPACE]Q"
810  w$(3,1)="Q[2xSPACE]":w$(3,2)="[SPA
     CE]Q[SPACE]":w$(3,3)="[2xSPACE]Q"
820  w$(4,1)="Q[SPACE]Q":w$(4,2)="[3xSP
     ACE]":w$(4,3)="Q[SPACE]Q"
830  w$(5,1)="Q[SPACE]Q":w$(5,2)="[SPAC
     E]Q[SPACE]":w$(5,3)="Q[SPACE]Q"
840  w$(6,1)="Q[SPACE]Q":w$(6,2)="Q[SPA
     CE]Q":w$(6,3)="Q[SPACE]Q"
850  print"[CRSR-DOWN][SPACE]'1-5'[SPAC
     E]=[SPACE]vasthouden[4xSPACE]'w'[S
     PACE]=[SPACE]werpen"
860  print"[2xSPACE]'p'[2xSPACE]=[SPACE
     ]punten[SPACE]tellen[SPACE]'v'[SPA
     CE]=[SPACE]veranderen
870  print"[COM-7][HOME][12xCRSR-DOWN][
     2xSPACE][COM-A][SHIFT-*][COM-R][13
     xSHIFT-*][COM-R][SHIFT-*][COM-R][1
     6xSHIFT-*][COM-S]"
880  print"[2xSPACE][SHIFT -]1[SHIFT -]
     een[4xSPACE]=[5xSPACE][SHIFT -]x[S
     HIFT -]3[SPACE]gelijk[2xSPACE]=[5x
     SPACE][SHIFT -]"
890  print"[2xSPACE][SHIFT -]2[SHIFT -]
     twee[3xSPACE]=[5xSPACE][SHIFT -]u[
     SHIFT—]4[SPACE]gelijk[2xSPACE]=[5
     xSPACE][SHIFT -]"
900  print"[2xSPACE][SHIFT -]3[SHIFT -]
     drie[3xSPACE]=[5xSPACE][SHIFT -]h[
     SHIFT—]full[SPACE]house=[5xSPACE]
     [SHIFT -]"
910  print"[2xSPACE][SHIFT -]4[SHIFT -]
     vier[3xSPACE]=[5xSPACE][SHIFT -]k[
     SHIFT—]kl.straat[SPACE]=[5xSPACE]
     [SHIFT -]"
920  print"[2xSPACE][SHIFT -]5[SHIFT -]
     vijf[3xSPACE]=[5xSPACE][SHIFT -]g[
     SHIFT—]gr.straat[SPACE]=[5xSPACE]
     [SHIFT -]"
930  print"[2xSPACE][SHIFT -]6[SHIFT -]
     zes[4xSPACE]=[5xSPACE][SHIFT -]y[S
     HIFT -]yahtzee[3xSPACE]=[5xSPACE][
     SHIFT—]"
940  print"[2xSPACE][SHIFT -]-[SHIFT -]
     bonus[2xSPACE]=[5xSPACE][SHIFT -]c
     [SHIFT -]chance[4xSPACE]=[5xSPACE]
     [SHIFT -]"
950  print"[2xSPACE][SHIFT -]-[SHIFT -]
     totaal[KWADRAAT PIJL=[5xSPACE][SHI
      FT -]-[SHIFT -]bonus[5xSPACE]=[5xS
      PACE][SHIFT -]"
960  print"[2xSPACE][COM-Z][SHIFT-*][CO
     M E][13xSHIFT-*][COM-E][SHIFT-*][C
     OM W]totaal[KWADRAAT PIJL][3xSPACE
      ]=[5xSPACE][SHIFT -]"
970  print"[20xCRSR-RIGHT][SHIFT -]punt
     en[4xSPACE]=[5xSPACE][SHIFT -]"
980  print"[20xCRSR-RIGHT][COM-Z][16xSH
     IFT *][COM-X]"
990  gosub1000:goto1110
1000 print"[COM-7][HOME][12xCRSR-DOWN]"
1010 printtab(3);ei$(n);tab(13);ei(n);t
     ab(19);p3$(n);tab(32);p3(n)
1020 printtab(3);zw$(n);tab(13);zw(n);t
     ab(19);p4$(n);tab(32);p4(n)
1030 printtab(3);dr$(n);tab(13);dr(n);t
     ab(19);fh$(n);tab(32);fh(n)
1040 printtab(3);vi$(n);tab(13);vi(n);t
     ab(19);ks$(n);tab(32);ks(n)
1050 printtab(3);fu$(n);tab(13);fu(n);t
     ab(19);gs$(n);tab(32);gs(n)
1060 printtab(3);se$(n);tab(13);se(n);t
     ab(19);ya$(n);tab(32);ya(n)
1070 printtab(3);bo$(n);tab(13);bo(n);t
     ab(19);ch$(n);tab(32);ch(n)
1080 printtab(3);ge$(n);tab(13);ge(n);t
     ab(19);bn$(n);tab(32);bn(n)
1090 printtab(19);gz$(n);tab(32);gz(n)
1100 printtab(19);pu$(n);tab(32);pu(n):
     return
1110 wu=3:h1=0:h2=0:h3=0:h4=0:h5=0
1120 forl=1to7
1130 ifh1=0thenw1=int(rnd(1)*6+1)
1140 ifh2=0thenw2=int(rnd(1)*6+1)
1150 ifh3=0thenw3=int(rnd(1)*6+1)
1160 ifh4=0thenw4=int(rnd(1)*6+1)
1170 ifh5=0thenw5=int(rnd(1)*6+1)
1180 ifh6=0thenw6=int(rnd(1)*6+1)
1190 print"[CTRL-1][HOME][2xCRSR-DOWN]"
     ;tab(4);w$(w1,1)
1200 printtab(4);w$(w1,2)
1210 printtab(4);w$(w1,3)
1220 print"[HOME][2xCRSR-DOWN]";tab(11)
     ;w$(w2,1)
1230 printtab(11);w$(w2,2)
1240 printtab(11);w$(w2,3)
1250 print"[HOME][2xCRSR-DOWN]";tab(18)
     ;w$(w3,1)
1260 printtab(18);w$(w3,2)
1270 printtab(18);w$(w3,3)
1280 print"[HOME][2xCRSR-DOWN]";tab(25)
     ;w$(w4,1)
1290 printtab(25);w$(w4,2)
1300 printtab(25);w$(w4,3)
1310 print"[HOME][2xCRSR-DOWN]";tab(32)
     ;w$(w5,1)
1320 printtab(32);w$(w5,2)
1330 printtab(32);w$(w5,3)
1340 next:wu=wu-1
1350 qq$="en":ifwu=1thenqq$=""
1360 print"[COM-6][HOME][10xCRSR-DOWN][
     14xCRSR-RIGHT]nog";wu;"worp"qq$
1370 ifwu=0then1510
1380 getq$:print"[HOME][8xCRSR-DOWN]";
1390 ifq$=""thenprint"[COM-6][CTRL-9][C
     OM *][CTRL-0][CRSR-DOWN][CRSR-LEFT
     ][SHIFT-|]":gosub3450:print"[2xCRS
     R-UP][SPACE][CRSR-LEFT][CRSR-DOWN]
     [SPACE][CRSR-LEFT]":gosub3450:goto1380
1400 gosub370
1410 ifq$="1"thenh1=1:print"[HOME][6xCR
     SR-DOWN]";tab(4);"[COM-1][CTRL-9][
     3xSPACE][CTRL-0]"
1420 ifq$="2"thenh2=1:print"[HOME][6xCR
     SR-DOWN]";tab(11);"[COM-1][CTRL-9]
     [3xSPACE][CTRL-0]"
1430 ifq$="3"thenh3=1:print"[HOME][6xCR
     SR-DOWN]";tab(18);"[COM-1][CTRL-9]
     [3xSPACE][CTRL-0]"
1440 ifq$="4"thenh4=1:print"[HOME][6xCR
     SR-DOWN]";tab(25);"[COM-1][CTRL-9]
     [3xSPACE][CTRL-0]"
```

```
1450 ifq$="5"thenh5=1:print"[HOME][6xCR
     SR-DOWN]";tab(32);"[COM-1][CTRL-9]
     [3xSPACE][CTRL-0]"
1460 ifq$="p"then1510
1470 ifq$="v"thenprint"[HOME][6xCRSR-DO
     WN][COM-8][5xSPACE]1[6xSPACE]2[6xS
     PACE]3[6xSPACE]4[6xSPACE]5[SPACE]"
     :goto1500
1480 ifq$="w"then1120
1490 goto1380
1500 h1=0:h2=0:h3=0:h4=0:h5=0:goto1380
1510 print"[CTRL-6][HOME][11xCRSR-DOWN]
     [3xCRSR-RIGHT]v[15xCRSR-RIGHT]v":g
     osub3450
1520 print"[CTRL-6][HOME][11xCRSR-DOWN]
     [3xCRSR-RIGHT][SPACE][15xCRSR-RIGH
     T][SPACE]"
1530 getq$:ifq$=""thengosub3450:goto1510
1540 gosub400
1550 ifei$(n)=""andq$="1"thenei$(n)="-"
     :goto1690
1560 ifp3$(n)=""andq$="x"thenp3$(n)="-"
     :goto2110
1570 ifzw$(n)=""andq$="2"thenzw$(n)="-"
     :goto1760
1580 ifp4$(n)=""andq$="u"thenp4$(n)="-"
     :goto2460
1590 ifdr$(n)=""andq$="3"thendr$(n)="-"
     :goto1830
1600 iffh$(n)=""andq$="h"thenfh$(n)="-"
     :goto2810
1610 ifvi$(n)=""andq$="4"thenvi$(n)="-"
     :goto1900
1620 ifks$(n)=""andq$="k"thenks$(n)="-"
     :goto2940
1630 iffu$(n)=""andq$="5"thenfu$(n)="-"
     :goto1970
1640 ifgs$(n)=""andq$="g"thengs$(n)="-"
     :goto3110
1650 ifse$(n)=""andq$="6"thense$(n)="-"
     :goto2040
1660 ifya$(n)=""andq$="y"thenya$(n)="-"
     :goto3170
1670 ifch$(n)=""andq$="c"thench$(n)="-"
     :goto3200
1680 goto1510
1690 rem een
1700 ifw1=1thenei(n)=ei(n)+1
1710 ifw2=1thenei(n)=ei(n)+1
1720 ifw3=1thenei(n)=ei(n)+1
1730 ifw4=1thenei(n)=ei(n)+1
1740 ifw5=1thenei(n)=ei(n)+1
1750 goto3460
1760 rem twee
1770 ifw1=2thenzw(n)=zw(n)+2
1780 ifw2=2thenzw(n)=zw(n)+2
1790 ifw3=2thenzw(n)=zw(n)+2
1800 ifw4=2thenzw(n)=zw(n)+2
1810 ifw5=2thenzw(n)=zw(n)+2
1820 goto3460
1830 rem drie
1840 ifw1=3thendr(n)=dr(n)+3
1850 ifw2=3thendr(n)=dr(n)+3
1860 ifw3=3thendr(n)=dr(n)+3
1870 ifw4=3thendr(n)=dr(n)+3
1880 ifw5=3thendr(n)=dr(n)+3
1890 goto3460
1900 rem vier
1910 ifw1=4thenvi(n)=vi(n)+4
1920 ifw2=4thenvi(n)=vi(n)+4
1930 ifw3=4thenvi(n)=vi(n)+4
1940 ifw4=4thenvi(n)=vi(n)+4
1950 ifw5=4thenvi(n)=vi(n)+4
1960 goto3460
1970 rem vijf
1980 ifw1=5thenfu(n)=fu(n)+5
1990 ifw2=5thenfu(n)=fu(n)+5
2000 ifw3=5thenfu(n)=fu(n)+5
2010 ifw4=5thenfu(n)=fu(n)+5
2020 ifw5=5thenfu(n)=fu(n)+5
2030 goto3460
2040 rem zes
2050 ifw1=6thense(n)=se(n)+6
2060 ifw2=6thense(n)=se(n)+6
2070 ifw3=6thense(n)=se(n)+6
2080 ifw4=6thense(n)=se(n)+6
2090 ifw5=6thense(n)=se(n)+6
2100 goto3460
2110 rem 3 gelijk
2120 v1=0:v2=0:v3=0:v4=0:v5=0:v6=0
2130 ifw1=1thenv1=v1+1
2140 ifw1=2thenv2=v2+1
2150 ifw1=3thenv3=v3+1
2160 ifw1=4thenv4=v4+1
2170 ifw1=5thenv5=v5+1
2180 ifw1=6thenv6=v6+1
2190 ifw2=1thenv1=v1+1
2200 ifw2=2thenv2=v2+1
2210 ifw2=3thenv3=v3+1
2220 ifw2=4thenv4=v4+1
2230 ifw2=5thenv5=v5+1
2240 ifw2=6thenv6=v6+1
2250 ifw3=1thenv1=v1+1
2260 ifw3=2thenv2=v2+1
2270 ifw3=3thenv3=v3+1
2280 ifw3=4thenv4=v4+1
2290 ifw3=5thenv5=v5+1
2300 ifw3=6thenv6=v6+1
2310 ifw4=1thenv1=v1+1
2320 ifw4=2thenv2=v2+1
2330 ifw4=3thenv3=v3+1
2340 ifw4=4thenv4=v4+1
2350 ifw4=5thenv5=v5+1
2360 ifw4=6thenv6=v6+1
2370 ifw5=1thenv1=v1+1
2380 ifw5=2thenv2=v2+1
2390 ifw5=3thenv3=v3+1
2400 ifw5=4thenv4=v4+1
2410 ifw5=5thenv5=v5+1
2420 ifw5=6thenv6=v6+1
2430  ifv1>=3orv2>=3orv3>=3orv4>=3orv5>=
     3orv6>=3then2450
2440 goto3460
2450 p3(n)=w1+w2+w3+w4+w5:goto3460
2460 rem 4 gelijk
2470 v1=0:v2=0:v3=0:v4=0:v5=0:v6=0
2480 ifw1=1thenv1=v1+1
2490 ifw1=2thenv2=v2+1
2500 ifw1=3thenv3=v3+1
2510 ifw1=4thenv4=v4+1
2520 ifw1=5thenv5=v5+1
2530 ifw1=6thenv6=v6+1
2540 ifw2=1thenv1=v1+1
2550 ifw2=2thenv2=v2+1
2560 ifw2=3thenv3=v3+1
2570 ifw2=4thenv4=v4+1
```

```
2580 ifw2=5thenv5=v5+1
2590 ifw2=6thenv6=v6+1
2600 ifw3=1thenv1=v1+1
2610 ifw3=2thenv2=v2+1
2620 ifw3=3thenv3=v3+1
2630 ifw3=4thenv4=v4+1
2640 ifw3=5thenv5=v5+1
2650 ifw3=6thenv6=v6+1
2660 ifw4=1thenv1=v1+1
2670 ifw4=2thenv2=v2+1
2680 ifw4=3thenv3=v3+1
2690 ifw4=4thenv4=v4+1
2700 ifw4=5thenv5=v5+1
2710 ifw4=6thenv6=v6+1
2720 ifw5=1thenv1=v1+1
2730 ifw5=2thenv2=v2+1
2740 ifw5=3thenv3=v3+1
2750 ifw5=4thenv4=v4+1
2760 ifw5=5thenv5=v5+1
2770 ifw5=6thenv6=v6+1
2780 ifv1>=4orv2> = 4orv3> = 4orv4> =
     4orv5> = 4orv6>=4then2800
2790 goto3460
2800 p4(n)=w1+w2+w3+w4+w5:goto3460
2810 rem foul house
2820 ifw1=w2andw2=w3andw4=w5then2930
2830 ifw2=w3andw3=w4andw1=w5then2930
2840 ifw3=w4andw4=w5andw1=w2then2930
2850 ifw1=w3andw3=w5andw2=w4then2930
2860 ifw1=w2andw2=w4andw3=w5then2930
2870 ifw2=w4andw4=w5andw1=w3then2930
2880 ifw2=w3andw3=w5andw1=w4then2930
2890 ifw1=w4andw4=w5andw2=w3then2930
2900 ifw1=w3andw3=w4andw2=w5then2930
2910 ifw1=w2andw2=w5andw3=w4then2930
2920 goto3460
2930 fh(n)=25:goto3460
2940 rem kleine straat
2950  if(w1+w2+w3+w4)=10and(w1<>w2<>w3<>
     w4)thenks(n)=30
2960  if(w2+w3+w4+w5)=10and(w2<>w3<>w4<>
     w5)thenks(n)=30
2970  if(w1+w3+w4+w5)=10and(w1<>w3<>w4<>
     w5)thenks(n)=30
2980  if(w1+w2+w4+w5)=10and(w1<>w2<>w4<>
     w5)thenks(n)=30
2990  if(w1+w2+w3+w5)=10and(w1<>w2<>w3<>
     w5)thenks(n)=30
3000  if(w1+w2+w3+w4)=14and(w1<>w2<>w3<>
     w4)thenks(n)=30
3010  if(w2+w3+w4+w5)=14and(w2<>w3<>w4<>
     w5)thenks(n)=30
3020  if(w1+w3+w4+w5)=14and(w1<>w3<>w4<>
     w5)thenks(n)=30
3030  if(w1+w2+w4+w5)=14and(w1<>w2<>w4<>
     w5)thenks(n)=30
3040  if(w1+w2+w3+w5)=14and(w1<>w2<>w3<>
     w5)thenks(n)=30
3050  if(w1+w2+w3+w4)=18and(w1<>w2<>w3<>
     w4)thenks(n)=30
3060  if(w2+w3+w4+w5)=18and(w2<>w3<>w4<>
     w5)thenks(n)=30
3070  if(w1+w3+w4+w5)=18and(w1<>w3<>w4<>
     w5)thenks(n)=30
3080  if(w1+w2+w4+w5)=18and(w1<>w2<>w4<>
     w5)thenks(n)=30
3090  if(w1+w2+w3+w5)=18and(w1<>w2<>w3<>
     w5)thenks(n)=30
3100 goto3460
3110 rem grote straat
3120 ifw1+w2+w3+w4+w5=20then3150
3130 ifw1+w2+w3+w4+w5=15then3150
3140 goto3460
3150  if(w1<>w2<>w3<>w4<>w5)thengs(n)=40
     :goto3460
3160 goto3460
3170 rem yahtzee
3180 ifw1=w2andw2=w3andw3=w4andw4=w5the
     nya(n)=50
3190 goto3460
3200 rem chance
3210 ch(n)=w1+w2+w3+w4+w5:goto3460
3220 print"[SHIFT-CLR][CTRL-4][12xCRSR-
     RIGHT]sp.1[2xSPACE]sp.2[2xSPACE]sp
     .3[2xSPACE]sp.4":print
3230 print"[CTRL-8]een[7xSPACE]=";tab(1
     2);ei(1);tab(18);ei(2);tab(24);ei(
     3);tab(30);ei(4)
3240 print"twee[6xSPACE]=";tab(12);zw(1
     );tab(18);zw(2);tab(24);zw(3);tab(
     30);zw(4)
3250 print"drie[6xSPACE]=";tab(12);dr(1
     );tab(18);dr(2);tab(24);dr(3);tab(
     30);dr(4)
3260 print"vier[6xSPACE]=";tab(12);vi(1
     );tab(18);vi(2);tab(24);vi(3);tab(
     30);vi(4)
3270 print"vijf[6xSPACE]=";tab(12);fu(1
     );tab(18);fu(2);tab(24);fu(3);tab(
     30);fu(4)
3280 print"zes[7xSPACE]=";tab(12);se(1)
     ;tab(18);se(2);tab(24);se(3);tab(3
     0);se(4)
3290 print"bonus[5xSPACE]=";tab(12);bo(
     1);tab(18);bo(2);tab(24);bo(3);tab
     (30);bo(4)
3300 print"totaal[4xSPACE]=";tab(12);ge
     (1);tab(18);ge(2);tab(24);ge(3);ta
     b(30);ge(4)
3310 print"3[SPACE]gelijk[2xSPACE]=";ta
     b(12);p3(1);tab(18);p3(2);tab(24);
     p3(3);tab(30);p3(4)
3320 print"4[SPACE]gelijk[2xSPACE]=";ta
     b(12);p4(1);tab(18);p4(2);tab(24);
     p4(3);tab(30);p4(4)
3330 print"full[SPACE]house=";tab(12);f
     h(1);tab(18);fh(2);tab(24);fh(3);t
     ab(30);fh(4)
3340 print"kl.[SPACE]straat=";tab(12);k
     s(1);tab(18);ks(2);tab(24);ks(3);t
     ab(30);ks(4)
3350 print"gr.[SPACE]straat=";tab(12);g
     s(1);tab(18);gs(2);tab(24);gs(3);t
     ab(30);gs(4)
3360 print"yahtzee[3xSPACE]=";tab(12);y
     a(1);tab(18);ya(2);tab(24);ya(3);t
     ab(30);ya(4)
3370 print"chance[4xSPACE]=";tab(12);ch
     (1);tab(18);ch(2);tab(24);ch(3);ta
     b(30);ch(4)
3380 print"totaal[4xSPACE]=";tab(12);gz
     (1);tab(18);gz(2);tab(24);gz(3);ta
     b(30);gz(4)
3390 print"bonus[5xSPACE]=";tab(12);bn(
     1);tab(18);bn(2);tab(24);bn(3);tab
     (30);bn(4)
```

```
3400 print"punten[4xSPACE]=";tab(12);pu
     (1);tab(18);pu(2);tab(24);pu(3);ta
     b(30);pu(4)
3410 print"[CRSR-DOWN][SPACE]nieuw[SPAC
     E]spel[SPACE]=[SPACE]'n'"
3420 getq$:ifq$=""then3420
3430 ifq$="n"thenrun
3440 end
3450 forv=1to50:next:return
3460 ge(n)=0:ge(n)=ge(n)+ei(n)+zw(n)+dr
     (n)+vi(n)+fu(n)+se(n)
3470 ifge(n)>=63thenge(n)=ge(n)+35:bo(n
     )=35
3480 gz(n)=0:gz(n)=gz(n)+p3(n)+p4(n)+fh
     (n)+ks(n)+gs(n)+ya(n)+ch(n)
3490 ifgz(n)>=100thengz(n)=gz(n)+100:bn
     (n)=100
3500 pu(n)=ge(n)+gz(n)
3510 gosub1000
3520 print"[COM-3][HOME][11xCRSR-DOWN][
     10xSPACE]v[SPACE]=[SPACE]volgende[
     SPACE]speler"
3530 gosub3450
3540 print"[HOME][11xCRSR-DOWN][30xSPACE]"
3550 gosub3450
3560 getq$:ifq$=""then3520
3570 gosub370:gosub400:gosub370:gosub330
3580 b=b+1:ifb=kthen3220
3590 n=n+1:ifn>sthenn=1
3600 print"[HOME][COM-5][23xCRSR-DOWN][
     19xSPACE]"
3610 print"[HOME][COM-5][22xCRSR-DOWN][
     4xCRSR-RIGHT]er[SPACE]speelt[SPACE
     ]nu[SPACE]:":print"[CTRL-4][6xCRSR
     -RIGHT]";sp$(n)
3620 print"[HOME][6xCRSR-DOWN][COM-8][5
     xSPACE]1[6xSPACE]2[6xSPACE]3[6xSPA
     CE]4[6xSPACE]5[SPACE]"
3630 print"[COM-6][HOME][10xCRSR-DOWN][
     14xCRSR-RIGHT]nog[SPACE]3[SPACE]wo
     rpen
3640 goto870
```

EINDE LISTING yathzee

checksum Yathzee

REGEL 10	26	REGEL 560	124	REGEL 1050	250	REGEL 1540	33
REGEL 20	59	REGEL 570	241	REGEL 1060	244	REGEL 1550	203
REGEL 30	163	REGEL 580	37	REGEL 1070	200	REGEL 1560	208
REGEL 40	59	REGEL 590	206	REGEL 1080	200	REGEL 1570	16
REGEL 50	201	REGEL 600	133	REGEL 1090	85	REGEL 1580	215
REGEL 60	59	REGEL 610	137	REGEL 1100	37	REGEL 1590	217
REGEL 70	129	REGEL 620	33	REGEL 1110	132	REGEL 1600	221
REGEL 80	59	REGEL 630	56	REGEL 1120	139	REGEL 1610	234
REGEL 90	12	REGEL 640	196	REGEL 1130	199	REGEL 1620	4
REGEL 100	16	REGEL 650	197	REGEL 1140	201	REGEL 1630	234
REGEL 110	127	REGEL 660	20	REGEL 1150	203	REGEL 1640	238
REGEL 120	16	REGEL 670	114	REGEL 1160	205	REGEL 1650	218
REGEL 130	239	REGEL 680	20	REGEL 1170	207	REGEL 1660	6
REGEL 200	49	REGEL 690	228	REGEL 1180	209	REGEL 1670	204
REGEL 210	205	REGEL 700	83	REGEL 1190	201	REGEL 1680	80
REGEL 220	242	REGEL 710	129	REGEL 1200	134	REGEL 1690	103
REGEL 230	118	REGEL 720	213	REGEL 1210	135	REGEL 1700	132
REGEL 240	197	REGEL 730	163	REGEL 1220	104	REGEL 1710	133
REGEL 250	207	REGEL 740	96	REGEL 1230	181	REGEL 1720	134
REGEL 260	192	REGEL 750	91	REGEL 1240	182	REGEL 1730	135
REGEL 270	91	REGEL 760	164	REGEL 1250	112	REGEL 1740	136
REGEL 280	71	REGEL 770	220	REGEL 1260	189	REGEL 1750	86
REGEL 290	107	REGEL 780	141	REGEL 1270	190	REGEL 1760	196
REGEL 300	91	REGEL 790	56	REGEL 1280	111	REGEL 1770	204
REGEL 310	73	REGEL 800	12	REGEL 1290	188	REGEL 1780	205
REGEL 320	175	REGEL 810	224	REGEL 1300	189	REGEL 1790	206
REGEL 330	203	REGEL 820	180	REGEL 1310	110	REGEL 1800	207
REGEL 340	91	REGEL 830	136	REGEL 1320	187	REGEL 1810	208
REGEL 350	73	REGEL 840	92	REGEL 1330	188	REGEL 1820	86
REGEL 360	29	REGEL 850	192	REGEL 1340	162	REGEL 1830	179
REGEL 370	112	REGEL 860	255	REGEL 1350	6	REGEL 1840	152
REGEL 380	68	REGEL 870	10	REGEL 1360	33	REGEL 1850	153
REGEL 390	29	REGEL 880	242	REGEL 1370	135	REGEL 1860	154
REGEL 400	109	REGEL 890	78	REGEL 1380	3	REGEL 1870	155
REGEL 410	68	REGEL 900	254	REGEL 1390	38	REGEL 1880	156
REGEL 420	29	REGEL 910	241	REGEL 1400	39	REGEL 1890	86
REGEL 430	204	REGEL 920	233	REGEL 1410	153	REGEL 1900	197
REGEL 440	77	REGEL 930	67	REGEL 1420	201	REGEL 1910	172
REGEL 450	77	REGEL 940	65	REGEL 1430	210	REGEL 1920	173
REGEL 460	229	REGEL 950	172	REGEL 1440	210	REGEL 1930	174
REGEL 470	62	REGEL 960	28	REGEL 1450	210	REGEL 1940	175
REGEL 480	229	REGEL 970	242	REGEL 1460	180	REGEL 1950	176
REGEL 490	62	REGEL 980	139	REGEL 1470	108	REGEL 1960	86
REGEL 500	63	REGEL 990	212	REGEL 1480	184	REGEL 1970	190
REGEL 510	168	REGEL 1000	86	REGEL 1490	85	REGEL 1980	166
REGEL 520	73	REGEL 1010	178	REGEL 1500	72	REGEL 1990	167
REGEL 530	128	REGEL 1020	250	REGEL 1510	18	REGEL 2000	168
REGEL 540	36	REGEL 1030	216	REGEL 1520	211	REGEL 2010	169
REGEL 550	204	REGEL 1040	10	REGEL 1530	208	REGEL 2020	170

vervolg checksum yathzee

REGEL 2030	86
REGEL 2040	129
REGEL 2050	162
REGEL 2060	163
REGEL 2070	164
REGEL 2080	165
REGEL 2090	166
REGEL 2100	86
REGEL 2110	120
REGEL 2120	167
REGEL 2130	56
REGEL 2140	59
REGEL 2150	62
REGEL 2160	65
REGEL 2170	68
REGEL 2180	71
REGEL 2190	57
REGEL 2200	60
REGEL 2210	63
REGEL 2220	66
REGEL 2230	69
REGEL 2240	72
REGEL 2250	58
REGEL 2260	61
REGEL 2270	64
REGEL 2280	67
REGEL 2290	70
REGEL 2300	73
REGEL 2310	59
REGEL 2320	62
REGEL 2330	65
REGEL 2340	68
REGEL 2350	71
REGEL 2360	74
REGEL 2370	60
REGEL 2380	63
REGEL 2390	66
REGEL 2400	69
REGEL 2410	72
REGEL 2420	75
REGEL 2430	42
REGEL 2440	86
REGEL 2450	190
REGEL 2460	121
REGEL 2470	167
REGEL 2480	56
REGEL 2490	59
REGEL 2500	62
REGEL 2510	65
REGEL 2520	68
REGEL 2530	71
REGEL 2540	57
REGEL 2550	60
REGEL 2560	63
REGEL 2570	66
REGEL 2580	69
REGEL 2590	72
REGEL 2600	58
REGEL 2610	61
REGEL 2620	64
REGEL 2630	67
REGEL 2640	70
REGEL 2650	73
REGEL 2660	59
REGEL 2670	62
REGEL 2680	65
REGEL 2690	68
REGEL 2700	71
REGEL 2710	74
REGEL 2720	60
REGEL 2730	63
REGEL 2740	66
REGEL 2750	69
REGEL 2760	72
REGEL 2770	75
REGEL 2780	47
REGEL 2790	86
REGEL 2800	191
REGEL 2810	73
REGEL 2820	175
REGEL 2830	176
REGEL 2840	177
REGEL 2850	176
REGEL 2860	175
REGEL 2870	177
REGEL 2880	176
REGEL 2890	177
REGEL 2900	176
REGEL 2910	175
REGEL 2920	86
REGEL 2930	214
REGEL 2940	22
REGEL 2950	94
REGEL 2960	102
REGEL 2970	100
REGEL 2980	98
REGEL 2990	96
REGEL 3000	98
REGEL 3010	106
REGEL 3020	104
REGEL 3030	102
REGEL 3040	100
REGEL 3050	102
REGEL 3060	110
REGEL 3070	108
REGEL 3080	106
REGEL 3090	104
REGEL 3100	86
REGEL 3110	223
REGEL 3120	105
REGEL 3130	109
REGEL 3140	86
REGEL 3150	164
REGEL 3160	86
REGEL 3170	169
REGEL 3180	167
REGEL 3190	86
REGEL 3200	49
REGEL 3210	198
REGEL 3220	76
REGEL 3230	115
REGEL 3240	190
REGEL 3250	65
REGEL 3260	119
REGEL 3270	96
REGEL 3280	23
REGEL 3290	144
REGEL 3300	186
REGEL 3310	186
REGEL 3320	191
REGEL 3330	180
REGEL 3340	209
REGEL 3350	195
REGEL 3360	71
REGEL 3370	147
REGEL 3380	14
REGEL 3390	140
REGEL 3400	51
REGEL 3410	131
REGEL 3420	182
REGEL 3430	117
REGEL 3440	128
REGEL 3450	71
REGEL 3460	172
REGEL 3470	199
REGEL 3480	127
REGEL 3490	127
REGEL 3500	11
REGEL 3510	78
REGEL 3520	243
REGEL 3530	89
REGEL 3540	171
REGEL 3550	89
REGEL 3560	183
REGEL 3570	64
REGEL 3580	131
REGEL 3590	24
REGEL 3600	15
REGEL 3610	184
REGEL 3620	240
REGEL 3630	153
REGEL 3640	4

14e Computer Info Beurs
3 november in de RAI
10.00-17.00

Toegang: *f* 7,50



BEELDMANIPULATIES (3)

Dit wordt de laatste aflevering met nieuwe theorie die echt nodig is. In volgende afleveringen zullen we steeds een onderwerp kiezen en dat bespreken. Maar deze keer moeten we nog even door wat harde theorie heen bijten.

Aan het einde van de eerste aflevering hebben we het er al even over gehad : de Klokcycli. Willen we de beeldmanipulaties op de Commodore 64 echt tot in de perfectie kunnen regelen dan zullen we precies moeten weten hoe een 'rasterlijn' wordt opgebouwd.

Klokcycli

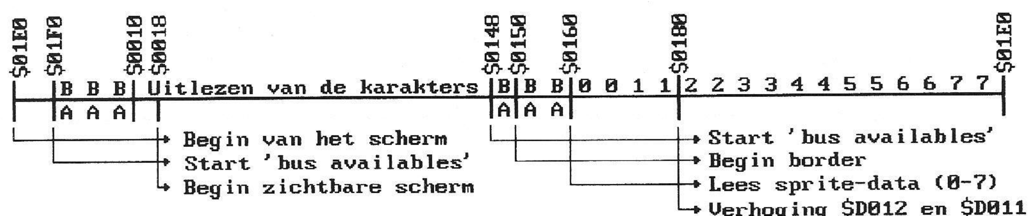
Zoals eerder gezegd is het scherm van de C-64 512 ($200) posities breed. Dit is alleen in theorie zo. In de praktijk blijkt dat er van de $200 posities er 8 niet op het scherm worden weer gegeven. Dit zijn de posities $1F8 tot $1FF. Je kunt de posities wel in het sprite register plaatsen maar er zal dan blijken dat ze niet op het scherm worden afgebeeld. Er zijn nog een aantal posities waar de sprites niet te zien zijn, dat zijn de posities vanaf $180 tot $1E0. Let wel, je kunt ze niet zien maar ze worden wel degelijk getekend op die posities. Als je dus een sprite op positie $1D0 plaatst en er zou geen border zijn (dit behandelen we later nog eens) dan zou de de sprite alleen te zien zijn vanaf positie $1E0 (alleen een stukje van de sprite dus).

Laten we eens een experiment doen. Verricht de volgende handelingen vanuit een machinetaal monitor:

```
1 F 2000 2040 FF
2 :D015 01
3 :07F8 80
4 :D01D 01
5 :D000 F8
6 :D001 A0
7 :D010 01
8 :D027 01
```

Even een korte uitleg. In regel 1 vullen we een stuk van $40 bytes op met de waarde $FF. In regel twee activeren we sprite 0. Regel drie zorgt er voor dat de commodore zijn gevens over de sprite van adres $2000 leest, dit zorgt er dus voor dat de sprite een vierkant is. Regel vier vergroot de sprite in de x richting. De regels 5 tot en met 7 zorgen er voor dat de sprite wordt neergezet op positie $1F8. En de laatste regel maakt de sprite wit waardoor hij beter te zijn is.

Na alles te hebben in getikt is er nergens op het scherm iets te zien. Maar als we nu intikken ':D000 F7' dan wordt opeens een sprite zichtbaar. Het enige wat we nu gedaan hebben is de x-positie 1 verminderen. Dit kleine stukje toont dus aan dat wanneer men een sprite op positie $1F8 plaats deze niet op het scherm wordt weergegeven.



Figuur 1

Nog even terug komen op de eerste aflevering. We hebben toen voor de duidelijkheid verteld dat een rasterregel 64 cycli was, immers 64*8 is $200. Maar na wat we hierboven hebben verteld blijkt dus dat we 8 posities minder hebben. Dit resulteert dus in $1F8/8 is 63 cycli voor één rasterregel.

Waarschijnlijk is tot jullie verbazing net gebleken dat de sprite op positie $1F7 aan de linkerkant van het scherm opdook, terwijl het logisch




```
SHOW-CYCLI

        *= $1000
;----------------

        JSR $E544  ;MAAK SCHERM SCHOON
        SEI
;----INITIALISEER INTERRUPT
        LDA #IRQ
        STA $0315
        LDA #
Q
        STA $0314
        LDX #$00
        STX $DC0E
        INX
        STX $D01A
        LDA #$32
        STA $D012
        LDA #$1B
        STA $D011
;----ZET TEKST EN NYBBLES OP SCHERM
        LDX #$05
        LDA #$A0
        STA $042A,X
        TXA
        STA $D82A,X
        DEX
        BPL *-10
        LDX #$06
        TXA
        CLC
        ADC #$30
        STA $0451,X
        DEX
        BNE *-8
;----PLAATS 256 NOPPEN NA DE INTERRUPT
        LDA #$EA
        LDX #$00
        STA NOPS,X
        INX
        BNE *-4
        CLI
        JMP NOPS
IRQ
        LDA #$01
        STA $D019

        BIT $2424
        BIT $2424
        BIT $2424
        BIT $2424
        BIT $2424
        BIT $2424

        DEC $D020
        LDA #14
        STA $D020

        BIT $2424
        BIT $2424
        BIT $2424
        BIT $2424

        LDA #$01
        STA $D021  ;MAAK SCHERM WIT
        LDA #$06
        STA $D021  ;MAAK WEER BLAUW

        JMP $EA81

NOPS               ;HIER BLIJFT DE
        *= *+256   ;ROUTINE IN EEN
        JMP NOPS   ;'LOOP' LOPEN
;TOTDAT EEN NIEUWE INTERRUPT ONTSTAAT
```

Listing 1

zou zijn te veronderstellen dat het ergens aan de rechter kant van het scherm had moeten zijn. Goed laten we nu dan de scherm opbouw van de commodore 64 eens nader bekijken. Zie **figuur 1** ter verduidelijking.

Zoals uit figuur 1 blijkt begint de C-64 helemaal aan de linker kant op positie $1E0 met het opbouwen van het scherm (lekker logisch toch?). Het zichtbare scherm begint pas op positie $0018. Dit is dus het punt vanwaar je de sprite helemaal kunt zien. Het einde van het zichtbare scherm vormt positie $150. Vanaf dat adres begint de commodore met het tekenen van de border. We gaan er hierbij vanuit dat het scherm in de normale stand staat. Je zou ook met de scroll registers $D011 en $D016 het zichtbare scherm met 8 posities kunnen verkleinen. Dit is bijvoorbeeld nodig bij het maken van een scroll (we komen hier in een volgende aflevering op terug).

De Commodore begint op positie $160 met het lezen van de data voor de sprite op de volgende rasterregel. Elke sprite die geactiveerd is vergt, op elke rasterregel waarop hij te zien is, 2 cycli om de data te lezen. Dit soort uitlezen van data door de VIC noemen we bus availables (BA's). Om deze data te kunnen lezen moet de VIC (de grafische chip) het geheugen in zijn eentje kunnen gebruiken. Om dit aan de processor duidelijk te maken heeft de VIC 3 cycli nodig voorafgaande aan het lezen van de data (deze handeling hebben we in het schema 'start BA's' genoemd, de VIC verzoekt om alleen toegang te krijgen tot de geheugen-bus). Deze cycli gaan dus vaak verloren bij het programmeren. Deze cycli worden als het ware gestolen van de programmeur. Dit vindt plaats op positie $148. Nog even duidelijkheid scheppen: Je bent deze 3 cycli per rasterregel alleen kwijt wanneer de sprite op de volgende rasterregel weergegeven wordt.

Elke achtste rasterregel moet de VIC ook opnieuw data lezen over de karakters die op het scherm moeten worden gezet. Immers een karakter is 8 rasterlijnen hoog. Hiervoor zijn weer 3 'start BA's' nodig en tevens kost het 40 cycli om alle data te lezen. Je bent dus elke achtste rasterregel 43 cycli kwijt ! Het uitlezen van de data begint op $0010, de data die op dat punt wordt gelezen wordt pas op $0018 weergeven. Zo loopt het le-



zen altijd voor op het weergeven. Dit verhaal gaat niet op wanneer de VIC de boven en onder border tekent. Hier zijn dus wel die 43 cycli te gebruiken.Uit het schema blijkt ook dat de posities $1F8 tot $1FF niet worden gebruikt. Als laatste, op positie $180 worden de adressen $D012 en $D011 verhoogt zodat zij de volgende rasterlijn aanwijzen.

Assembler instructies en hun cycli

Om beeldmanipulaties perfect te kunnen programmeren moeten we alles perfect 'timen'. Hiermee bedoelen we dat je dus zelf precies kunt bepalen op welk moment er iets op een rasterregel moet gebeuren. Hierbij is het natuurlijk nodig om te weten hoeveel tijd de computer nodig heeft om een bepaalde instructie uit te voeren. In figuur 2 is uit te lezen hoeveel cycli de computer nodig heeft voor een bepaalde instruktie. Even een voorbeeld. Neem de volgende instruktie sequence:

```
LDA #$01
STA $D020
```

Aan de hand van de tabel kunnen we nu berekenen dat deze twee instrukties bij elkaar opgeteld 6 cycli van de computer vergen. De LDA #$01 kost twee cycli en de STA $D020 kost de computer 4 cycli.

Bij de eerder genoemde BA's wordt door de computer gestopt met het uitvoeren van een instruktie. Na het lezen van de data gaat de computer dan weer verder met deze instruktie. Er zijn hierop een aantal uitzonderingen. Dit zijn instructies die de computer wel afmaakt tijdens de start BA's, maar deze worden alleen afgemaakt als de laatste 2 cycli van de instruktie in die start BA's vallen. Hiermee maken we dus 2 cycli winst op zo'n rasterregel. Want we voeren bijvoorbeeld een instructie uit van 6 cycli en daarvan worden er 4 cycli uitgevoerd in de voor ons beschikbare tijd en de andere twee worden door de VIC meegenomen. De instrukties die hiervoor in aanmerking komen zijn : INC, DEC, ASL, LSR, ROL, ROR en de BRK instruktie. Bij de BRK instruktie worden zelfs de laatste 3 cycli in de start BA's afgemaakt. Dit laatste heeft niet echt veel nut want we zullen niet zo maar een BRK midden in onze code plaatsen.

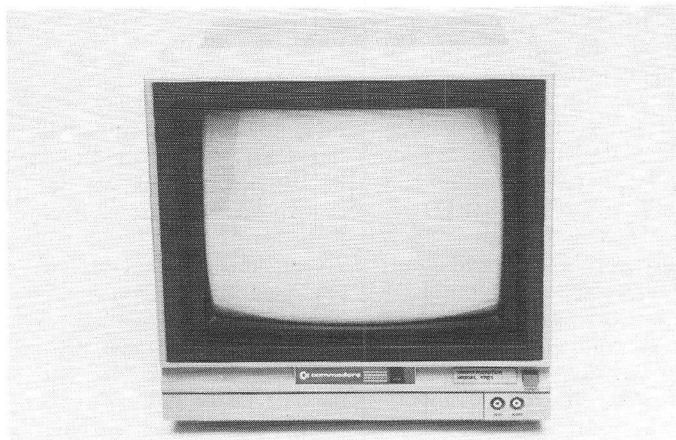
Er zijn na aanleiding van de tabel nog een paar dingen uit te leggen. Bijvoorbeeld over de 'Branch' instrukties. Een branch kost normaal 2 cycli maar als de branch over een zogenaamde 'page-boundary' gaat heet dit een 'pagebreak'. Elke $100 bytes is een pagina. Dus vanaf adres $1000 tot $10FF is één pagina. Als de branch over een pagina-grens heen komt heeft de instruktie één cycli meer nodig. Als de branch wordt uitgevoerd kost het ook een cycli extra. Een branch kan dus maximaal 4 cycli lang worden. Deze informatie geldt niet alleen voor de branch instruktie maar voor alle instrukties met een sterretje erbij.

INSTRUCTION ADDRESSING MODES AND RELATED EXECUTION TIMES (in clock cycles)

	Accumulator	Immediate	Zero Page	Zero Page, X	Zero Page, Y	Absolute	Absolute, X	Absolute, Y	Implied	Relative	(Indirect, X)	(Indirect), Y	Absolute Indirect
ADC	.	2	3	4	.	4	4*	4*	.	.	6	5*	.
AND	.	2	3	4	.	4	4*	4*	.	.	6	5*	.
ASL	2	.	5	6	.	6	7	.	.	.	.	.	.
BCC	.	.	.	.	.	.	.	.	.	2**	.	.	.
BCS	.	.	.	.	.	.	.	.	.	2**	.	.	.
BEQ	.	.	.	.	.	.	.	.	.	2**	.	.	.
BIT	.	.	3	.	.	4	.	.	.	.	.	.	.
BMI	.	.	.	.	.	.	.	.	.	2**	.	.	.
BNE	.	.	.	.	.	.	.	.	.	2**	.	.	.
BPL	.	.	.	.	.	.	.	.	.	2**	.	.	.
BRK	.	.	.	.	.	.	.	.	.	.	.	.	.
BVC	.	.	.	.	.	.	.	.	.	2**	.	.	.
BVS	.	.	.	.	.	.	.	.	.	2**	.	.	.
CLC	.	.	.	.	.	.	.	.	2	.	.	.	.
CLD	.	.	.	.	.	.	.	.	2	.	.	.	.
CLI	.	.	.	.	.	.	.	.	2	.	.	.	.
CLV	.	.	.	.	.	.	.	.	2	.	.	.	.
CMP	.	2	3	4	.	4	4*	4*	.	.	6	5*	.
CPX	.	2	3	.	.	4	.	.	.	.	.	.	.
CPY	.	2	3	.	.	4	.	.	.	.	.	.	.
DEC	.	.	5	6	.	6	7	.	.	.	.	.	.
DEX	.	.	.	.	.	.	.	.	2	.	.	.	.
DEY	.	.	.	.	.	.	.	.	2	.	.	.	.
EOR	.	2	3	4	.	4	4*	4*	.	.	6	5*	.
INC	.	.	5	6	.	6	7	.	.	.	.	.	.
INX	.	.	.	.	.	.	.	.	2	.	.	.	.
INY	.	.	.	.	.	.	.	.	2	.	.	.	.
JMP	.	.	.	.	.	3	.	.	.	.	.	.	5
JSR	.	.	.	.	.	6	.	.	.	.	.	.	.
LDA	.	2	3	4	.	4	4*	4*	.	.	6	5*	.
LDX	.	2	3	.	4	4	.	4*	.	.	.	.	.
LDY	.	2	3	4	.	4	4*	.	.	.	.	.	.
LSR	2	.	5	6	.	6	7	.	.	.	.	.	.
NOP	.	.	.	.	.	.	.	.	2	.	.	.	.
ORA	.	2	3	4	.	4	4*	4*	.	.	6	5*	.
PHA	.	.	.	.	.	.	.	.	3	.	.	.	.
PHP	.	.	.	.	.	.	.	.	3	.	.	.	.
PLA	.	.	.	.	.	.	.	.	4	.	.	.	.
PLP	.	.	.	.	.	.	.	.	4	.	.	.	.
ROL	2	.	5	6	.	6	7	.	.	.	.	.	.
ROR	2	.	5	6	.	6	7	.	.	.	.	.	.
RTI	.	.	.	.	.	.	.	.	6	.	.	.	.
RTS	.	.	.	.	.	.	.	.	6	.	.	.	.
SBC	.	2	3	4	.	4	4*	4*	.	.	6	5*	.
SEC	.	.	.	.	.	.	.	.	2	.	.	.	.
SED	.	.	.	.	.	.	.	.	2	.	.	.	.
SEI	.	.	.	.	.	.	.	.	2	.	.	.	.
STA	.	.	3	4	.	4	5	5	.	.	6	6	.
STX	.	.	3	.	4	4	.	.	.	.	.	.	.
STY	.	.	3	4	.	4	.	.	.	.	.	.	.
TAX	.	.	.	.	.	.	.	.	2	.	.	.	.
TAY	.	.	.	.	.	.	.	.	2	.	.	.	.
TSX	.	.	.	.	.	.	.	.	2	.	.	.	.
TXA	.	.	.	.	.	.	.	.	2	.	.	.	.
TXS	.	.	.	.	.	.	.	.	2	.	.	.	.
TYA	.	.	.	.	.	.	.	.	2	.	.	.	.

* Add one cycle if indexing across page boundary
** Add one cycle if branch is taken, Add one additional if branching operation crosses page boundary

© Commodore 64 Reference Guide



Laten we nu aan de hand van een listing een voorbeeld geven van het begrip cycli en de tijd die een instructie kost.

Als je dit programma ingetypt hebt en opstart zie je als het goed is zes verschillende kleuren blokjes (nybbles) met daaronder de getallen 1 t/m 6. Boven de nybbles staat een wit lijntje. Dit lijntje is precies even breed als de zes nybbles samen. Het gaat in dit voorbeeld voornamelijk om dat lijntje. In het programma wordt aan het einde van de interruptroutine het scherm wit gemaakt. Direkt daarna zetten we het scherm weer terug op z'n originele kleur. De tijd die dat kost komt overeen met de lengte van het witte lijntje. De tijd die het in deze routine kost is 6x8 puntjes en dat komt overeen met 6 cycli (1 cycli is 8 scherm puntjes). We gebruiken in het programmatje een LDA en een STA instruktie om de originele kleur te herstellen. Al eerder deze aflevering heb je kunnen lezen dat deze twee instrukties 6 cycli kost. Merk dus op dat de lengte van de witte lijn overeenkomt met het aantal cycli dat nodig is om de kleur weer goed te zetten.
In de interrupt routine komt een aantal keren het kommando BIT $2424 voor. Deze instruktie heeft geen direct nut, hij functioneert alleen maar als vertraging om dingen precies te 'timen'. Elk ander kommando moet namelijk precies op het juiste tijdstip plaatsvinden.

Op het moment dat een interruptsignaal gecreëerd wordt kan de processor net klaar zijn met een nop of zal hem afmaken (je kan namelijk zien dat buiten de interrupt een 'loop' van 256 noppen doorlopen wordt). Dit kan maximaal 2 cycli duren (zie tabel 2). Als er andere instrukties buiten de interrupt hadden gestaan die langer duren dan 2 cycli dan kan het langer dan 2 cycli duren voordat met de interruptroutine begonnen kan worden. De kans dat een interruptsignaal ontstaat als hij met de JMP bezig is (deze kan maximaal 4 cycli duren) is zo goed als 0%. Als het inderdaad gebeurt dat de processor eerst de nop instruktie afmaakt dan betekent dat dat de interruptroutine 2 cycli later begint.
Onder normale omstandigheden zou het witte lijntje in ons voorbeeldprogrammatje steeds 16 puntjes verschuiven, maar het bijzondere van dit programmatje is echter dat er een DEC $D020 in staat die precies 6 cycli duurt voordat start BA's gaan optreden wordt uitgevoerd, in het geval dat de processor net aan een nieuwe nop moet beginnen als de interrupt optreedt. Moet de processor de nop nog afmaken, dan start de routine 2 cycli later waardoor de laatste twee cycli van de DEC $D020 in de start BA's vallen. Zoals eerder vermeld maakt de processor dan deze opdracht af in de tijd van de start BA's. Op deze manier winnen we dus de verloren 2 cycli weer terug en loopt de routine synchroon met de andere mogelijkheid (dus de mogelijkheid dat de routine niet die 2 cycli verloren heeft), met als gevolg dat het lijntje niet meer die 16 puntjes verschuift. Was de opdracht buiten de interrupt langer geweest dan 2 cycli dan konden de laatste twee cycli van de DEC $D020 niet afgemaakt worden tijdens de start BA's.
Overigens de reden dat op de plek na de DEC $D020 start BA's staan is dat op die rasterregel data voor de letters moet worden ingelezen. Zoals je waarschijnlijk zelf ook al had kunnen uitvogelen wordt op rasterregel $33 de data voor de eerste regel ingelezen. Elke volgende 8 rasterregels herhaalt dit zich (totdat alle regels uitgelezen zijn) zoals ook al eerder vermeld is.
De slimmerikken onder ons zullen zich nu wel afvragen hoe het nu mogelijk is om en een constante rasterinterrupt en nog een programma buiten deze interrupt te kunnen laten lopen zonder dat de interrupt last heeft van de onregelmatigheden van het hoofdprogramma. Het antwoord is jammergenoeg niet zo eenvoudig want om dat op dezelfde methode aan te pakken zoals we hier gedaan hebben is zeer omslachtig en zal uiteindelijk veel te veel rastertijd gaan kosten. Een effectieve oplossing is gelukkig voor de komende afleveringen nog niet nodig, maar we komen er nog op terug. In de volgende aflevering zullen we ons voornamelijk bezighouden met de aantal verschillende manieren van tekstscrolling.

Wytze Westra &
Ansgar Smith



HYDRA ETHERNET

Een echt LAN voor de Amiga

De gedachte aan een echt Amiga-netwerk in de huiskamer of club zal hobbyisten niet echt aanspreken. Bij zakelijk gebruik kan een LAN wel degelijk een aantrekkelijke optie zijn. Populaire Amiga-toepassingen zoals desktoppublishing, desktopvideo, plaatjesdatabases, CAD-simulaties en koppeling aan een TV-netwerk varen wel in een netwerkstructuur. Meerdere gebruikers kunnen als groep aan één bepaald project werken (groupware) of dure randapparatuur gezamenlijk gebruikt worden. De juiste kaart in de Amiga en de wereld van datacommunicatie ligt aan uw voeten.

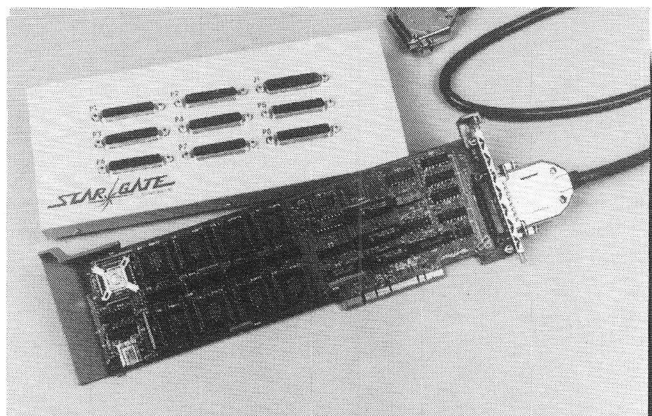
Datacommunicatie neemt hand over hand toe. Wie zegt dat dit bij de Amiga niet zo is heeft buiten de modemwaard gerekend. Menige Amiga-hobbyist heeft al de telefoonrekening flink laten oplopen door bulletin boards te raadplegen en met 'medefreaks' te communiceren.
Een netwerk is echter een heel ander kapittel. Dat leek aanvankelijk alleen iets voor IBM-compatibele PC's en UNIX-werkstations. Voor de Amiga was dat allemaal niet weggelegd. Daar is onlangs flink verandering in gekomen. Commodore zelf bracht de UNIX-compatibele en als fileserver bruikbare Amiga 3000 uit. Ook het gebrek aan netwerk-hardware is inmiddels geslecht. De eerste netwerkkaarten betreden, zij het wat schuchter, de netwerkarena voor de Amiga 500, 100, 2000 en 2500. Bijvoorbeeld de hier besproken Hydra Ethernetkaart. Praktisch gezien zal de netwerkkaart voornamelijk in de modellen 2000 en 2500 gestopt worden maar een connectie met de beide andere telgen is in principe goed mogelijk.
De belangrijkste redenen om met een netwerk te beginnen zijn een hogere efficiency en databescherming. Die efficiency bestrijkt een groot aantal terreinen, variërend van gezamenlijk gebruik van krachtige printer of superharddisks/CD-ROM's tot koppeling aan complete zetsystemen (DTP), telefoonnetwerken (PBX), videostudio's (DTV) en TV-circuits. In de industrie komt veel netwerkmatige procesbesturing voor.
De databescherming is een geheel andere invalshoek. Privacy- of bedrijfsveiligheid-gevoelige informatie kan keurig centraal worden opgeborgen en slingert niet meer rond op allerlei losse computers. Met behulp van wachtwoorden en privileges kan de netwerkbeheerder (sysop) de toegang tot en het gebruik van data en software reguleren.

Het LAN

Een van de vele Local Area Networksystemen (LAN) is Ethernet. Bij een LAN is er sprake van een netwerk tussen computers al of niet in combinatie met telefoon- of TV-circuits. Het kan om enkele maar ook vele tientallen aansluitingen gaan. In de praktijk slaat het 'Local' op een geografische beperking. Het netwerk reikt niet verder dan het kantoorgebouw, het industrieterrein om de stadsgrenzen. Omvat het netwerk een hele provincie, een heel land of meerdere landen dan spreekt men van een Wide Area Network of WAN.
Een LAN wordt doorgaans gebruikt voor:

° Informatie-distributie; bijvoorbeeld een centrale data-opslag (fileserver) waar de aangesloten werkstations informatie naar toe kunnen sturen en weer kunnen opvragen;
° Elektronische postvoorziening; electronic mail;
° Tekstnetwerken; zetterijen en desktoppublishing;
° CAD/CAM; groepsmatig (groupware design) ontwerpen;
° TV- of videotoepassingen variërend van het eenvoudig overbrengen van beelden tot simultaan animaties;
° Het doorverbinden van meerdere informatiebronnen;





- Het met meerdere computers gezamenlijk kunnen exploiteren van dure of specialistische randapparatuur;
- Multitasking; de stationgebruiker kan zware of langdurige taken aan het netwerk uitbesteden en zelf met iets anders verder gaan.

De laatste jaren is een vergaande integratie met de digitale bedrijfstelefooncentrale (PBX) en TV-circuits. Daarmee wordt het LAN tot een universeel datamedium dat zowel teksten, cijfers, grafische beelden, TV-beelden, als spraak en geluid kan transporteren.

Ethernet

Ethernet is een bepaalde LAN-standaard die het met name in de UNIX-wereld aardig doet. Een dergelijke netwerkstandaard beschrijft de communicatieprotocollen, de verbindingen, het softwarematige onderhoud van het netwerk en de wijze van toegang tot het LAN.
De grondleggers van Ethernet zijn DEC, Xerox en Intel. Zij legden een netwerkstandaard met een éénkanaals transmissie, coaxiaalkabelverbindingen, Frequency Shift Keying als modulatiemethode (= codering) en een overdrachtssnelheid van 10 Megabits. Later kwam de met goedkopere kabels werkende Cheapernetversie uit. Wie het geld er voor over heeft kan met Ethernet meer dan 8.000 computers aan elkaar koppelen.
In een Ethernet-netwerk treft u de volgende componenten aan:

- De coaxiaalkabel van 50 Ohm CATV (Let op: geen 75 standaard TV-coaxiaal);
- De transceiver met bijbehorende kabels koppelt de computerhardware aan het kabelmedium;
- De controllerkaart;
- De data-encoder/decoder;
- De repeaters voor het onderling verbinden van netwerksegmenten zodat een grote reikwijdte mogelijk is. N.B.: Een repeater zit altijd tussen twee transceivers;
- De aangesloten terminals (de PC"s of Amiga's).
- Verder behoort bij Ethernet de besturingssoftware die de timing, de transportfaciliteiten en het detecteren van fouten afhandelt.

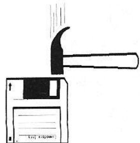
De Hydra Ethernet-kaart

Een netwerkkaart zoals de Hydra Ethernet-kaart biedt elke aangesloten Amiga een eigen stukje netwerkomgeving aan. In dit stukje afgeschermde netwerkomgeving waant de Amiga zich koning en denkt uitsluitend alleenheerser te zijn over alle op het LAN aangesloten randapparatuur. Lastpak Amiga DOS 1.3 kan dan geen roet meer in het datacommunicatie-eten gooien. De Ethernetkaart zorgt ervoor dat alle communicatie in goede banen wordt geleid.
De lange Hydra Ethernetkaart ziet er gewoontjes uit. Het enige dat opvalt zijn de beide aansluitingsbussen voor coaxiaalkabels aan de vleugelzijde en het rode dipswitch-blokje met bovenaan Ethernet en onderaan Cheapernet. Voor de rest zou het om een doorsnee modemkaart kunnen gaan.
De Hydrakaart wordt gewoon in het meest rechtse 100-polige Amiga-uitbreidingsslot gedrukt. Via een rood dipswitch-blokje kan de gebruiker de keuze maken uit de duurdere Ethernet- of de goedkopere en wat langzamere Cheapernet-aansluiting. Zit de kaart eenmaal in het slot dan kan de elektronica via DMA rechtstreeks op het lokale Amiga-geheugen aangrijpen. De Direct Memory Access kan maximaal 16 MB adresseren. Hoe meer RAM-geheugen, des te sneller de communicatie loopt. Minimaal 2 MB aan vrij RAM valt in deze aan te bevelen. Andere snelheidsbevorderende factoren zijn uiteraard een Motorola 68020/30 processor en een snelle harddisk.
De softwarematige installatie verloopt via STARTNET, REMOTEMOUNT en REMOTERUN. Geen onbekende DOS-termen voor de Amigagebruiker die al eens iets via de CLI in zijn of haar machine geïnstalleerd heeft.
Het kleinste netwerk bestaat uit twee Amiga's. Interessanter zijn LAN's met vier tot zes of nog meer Amiga's en Ethernetverbindingen met niet-Amigacomputers. UNIX en X-windows bieden deze mogelijkheid in ruime mate. Het is zelfs mogelijk via de Hydrakaart de Amiga als cliënt in een Novell-netwerk te laten opnemen.
Kortom, onbegrensde mogelijkheden voor wie met de Amiga op data-safari wil. Vanzelfsprekend moet je er wel een praktische toepassing voor hebben want de Ethernetinstallatie komt op ruim *f* 1.650,- per Amigastation.

Het verschijnen van Ethernet op de Amiga is weer een bewijs dat de Amiga in het PC-gebeuren meer en meer een professionele rol gaat spelen. Alleen lijkt ons de markt voor alsnog klein. Naar DTP'ers, DTV'ers, CAD/CAM'ers en wetenschappers zal de Hydra kaart zijn weg wel vinden. Voor de overige gebruikers is het nut twijfelachtig en zijn de investeringen niet gering.

Inl: 3Gitaal, 020-970035

U.S.

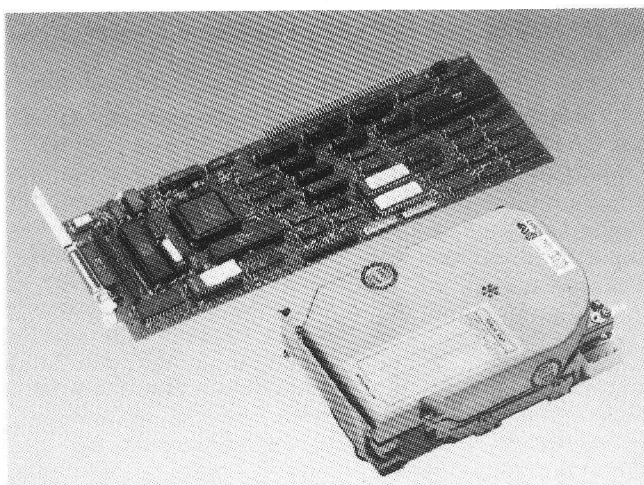


SCSI harddisk-controllers

16-bits power

Elke harddisk die u in of buiten de Amiga installeert heeft een of andere vorm van een controllerkaart nodig. Deze besturingskaart zit bij de Amiga 2000- en 2500-modellen in de systeemkast en bij externe Amiga 500-drives in de eigen drive behuizing. Bij de 2000-serie is er momenteel keuze uit twee verschillende interface-methoden voor harddisks: ST506/412 of de daarvan afgeleide ESDI, beiden 8-bits en de snellere 16-bits SCSI. Dit artikel behandelt de onderlinge verschillen. De keten is net zo sterk als de zwakste schakel. Dit gezegde vormt voor harddisks geen uitzondering. Te trage harde schijven of 8-bits controller-communicatie kunnen het systeem belemmeren. En dat is jammer, want de Amiga is tot zeer snelle datacommunicatie tussen 680X0 CPU en harde schijf in staat.

Het meest gebruikte harddisk-interfacetype is de door Seagate ontwikkelde 8-bits ST506. Momenteel is een kentering waarneembaar van langzaam oprukkende 16-bits SCSI-controllers. Een aantal controllerkaarten biedt zelfs beide mogelijkheden zodat de gebruiker met twee verschillende harde schijven in de Amiga kan werken.
In dit artikel zullen we eerst de oude standaards bekijken en daar vervolgens de voordelen van de SCSI-techniek tegen uitzetten.



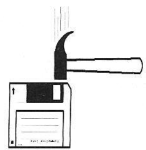
Commodore harddiskcontroller

De ST506-norm

Computerpioniers herinneren zich wellicht nog de 5 MegaByte Seagate harddisks. Deze voorloper van de moderne PC-harddisk kreeg van de fabriek een **ST506-controllerkaart** mee. Dit concept kreeg vlug navolging van andere fabrikanten en wordt nog steeds vaak bij de aansturing van harde schijven met vele tientallen MegaBytes aan opslagcapaciteit gebruikt. In de tussentijd bracht Seagate ook nog een **ST412-controller** voor 10 MB harddisks uit. De ST412 is in de praktijk volledig compatibel met de ST506 zodat de beide normen synoniem gebruikt kunnen worden.
De moderne uitvoering van de ST506 kan met twee opslagmethoden werken:

- **MFM**; Modified Frequency Modulation. Met frequentie modulatie geschreven en gelezen schijven hebben een kleinere opslagcapaciteit dan het hierna besproken RLL-type. De overdrachtssnelheid varieert van 250 tot 500 Kilobit per seconde.
- Op een **RLL 2.7** (Run Length Limited) harddisk kan het systeem ongeveer 50% informatie meer kwijt op hetzelfde schijfoppervlak. De overdrachtssnelheid ligt maximaal tegen de 800 Kilobit.

MFM-harddisks zijn nog altijd populairder dan hun RLL-collega's. Dat komt omdat de RLL-lees/schrijfwijze hogere nauwkeurigheids- en media-eisen stelt. Dat maakt de mechanische constructie en het schijfoppervlak duurder. Wie een gewone MFM harddisk op een RLL- controller probeert aan te sluiten kan rampspoed op zijn of haar pad verwachten. Data raken zoek of de schijf werkt niet naar behoren.
In opkomst is een nieuwe RLL-norm. De zogenaamde **ARLL** (Advanced Run Length Limited) of RLL 3.9. Het "advanced" slaat op een ruwweg dubbele opslagcapaciteit en een twee maal zo snelle toegangstijd. Uiteraard hangt er aan de benodigde nauwkeurigheid en media-specificaties een fors prijskaartje.
En ST506-achtige controller kenmerkt zich door twee **34-polige** aansluitingen. De ene connector is voor de bestu-



ring (met massa- en signaalbekabeling) en de ander voor de gegevensoverdracht (alleen signaalgeleiding). Het aansluiten van een dergelijke controller is een fluitje van een cent. Gewoon in het juiste uitbreidingsslot drukken, vleugel vastdraaien en lintkabeltje volgens de in de handleiding opgegeven oriëntatie aansluiten.
Zoals gezegd is de 8-bits ST506-controller technisch achterhaald. De voornaamste redenen voor de populariteit zijn de lage prijs en relatief hoge betrouwbaarheid. Een vertrouwd concept gooit men nu eenmaal niet zo gauw weg. Vele Amiga-controllers zijn nog op een ST506-interface gebaseerd.

De ESDI-norm

Het technisch modernere broertje van de ST506 is de **ESDI- controller**. ESDI staat voor Enhanced Small Device Interface. Er is een aantal wezenlijke verschillen met de ST506-norm:

- De controller verstuurt uitsluitend **digitale signalen** naar de harddisk-elektronica. In de ST506 zit een A/D-converter die de analoge schijfsignalen omzet in digitale PC-signalen;
- De controller is **aanzienlijk sneller** dan de oude ST506-norm. Een overdrachtssnelheid van 15 Megabit per seconde vormt geen uitzondering;
- De digitale schrijfwijze maakt de ESDI-harddisk **qua data-opslag betrouwbaarder**;
- Het **schijfformat zet 34 sectoren per spoor** tegen 17 sectoren per track bij ST506-drives neer;
- De controller kan ook **andere randapparatuur** dan harde schijven aansturen.

Door het ontbreken van de A/D-converter op de controller en de andere wijze van formatteren **kunnen ST506-drives dus niet op een ESDI-controller gebruikt worden!** Informeer daarom terdege bij de leverancier om wat voor type harddisk het gaat of koop beide in combinatie. Inmiddels zijn er diverse met de Commodore Amiga-lijn compatibele ESDI-controllers verkrijgbaar.

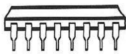
De SCSI-norm

Bij de Amiga en de PC is de **SCSI-controller** langzaam maar zeker in opkomst. SCSI staat voor Small Computer System Interface. Een belangrijk verschil met de hiervoor besproken controllers is de eigenschap van **System Interface**. Het interface werd van meet af aan ontworpen om allerlei typen randapparatuur aan te sturen en de harddisk stond niet primair aan de wieg van het ontwerp.
Een SCSI kan dus scanners, harddisks, tapestreamers, diskdrives faxen enzovoort aansturen. Bovendien kan men met één SCSI-controller zeven tot acht randapparaten aansturen.
Een ander belangrijk verschil is de **bloksgewijze dataoverdracht**. Alle informatie wordt in compacte blokken verpakt en met kleine tussenpozen verzonden. De beide andere genoemde controllers sturen de data teken voor teken over de lintkabel.
Wat de overdrachtssnelheid betreft zijn er twee mogelijkheden:

- **Synchrone overdracht** met maximaal 4 Megabit per seconde;
- **Asynchrone overdracht** met maximaal 1,5 Megabit per seconde.

In de praktijk ligt de snelheidslimiet ongeveer de helft lager.
Ook verschilt het **schijfformat**. Een SCSI-interface formatteert op 26 of 36 sectoren per track. Een hoge kwaliteit van de magnetische laag en drivenauwkeurigheid zijn weer een must.
Wat de Amiga betreft is eveneens de **16-bits busoverdracht** van belang. Alleen al die 16-bits connector is al twee maal zo snel als de voorgenoemde 8-bits controller-interfaces. Verder toont de SCSI-controller zich vrij tolerant tegen de aanwezigheid van een andere (ook van type) controller in het systeem. Dat kan van ST506- en ESDI-kaarten niet gezegd worden.
Tot zover slechts voordelen van de SCSI-controller. Zijn er ook nadelen? De enige echte nadelen zijn volgens ons de iets duurdere (25 tot 50%) aanschafprijs en de relatieve schaarste aan SCSI-compatibele randapparatuur. Voor de rest verdient zo'n krachtige machine als de Amiga gewoon een SCSI-controller!
Momenteel ziin SCSI-compatibele harddisk-controllers al of niet in combinatie met een drive-unit verkrijgbaar van Commodore (A2091), Great Valley Product (GVP Impact), Kronos en Golem. Vraag er naar in de goed voorziene computershop. Reken op circa *f* 2.200,- voor een 40 MB harddisk met SCSI-controller.

U.S.



8-UP!

Voordelige RAM-kaart van MicroBotics

In een Amiga 2000 is er eigenlijk altijd tekort aan vrij RAM. Slechts één MB is lang niet voldoende voor geheugenvretende HI-RES-graphics, digitizing en animaties. Wie het summum uit zijn of haar Amiga wil halen zal in extra RAM moeten investeren. In de wandelgangen hoor je dan als ondernemend Amiga-bezitter over moeilijke installaties en prijzige kaarten. Dat dat allemaal best meevalt bewijst de hier geteste 8-UP! (DIP) RAM-kaart van MicroBotics.

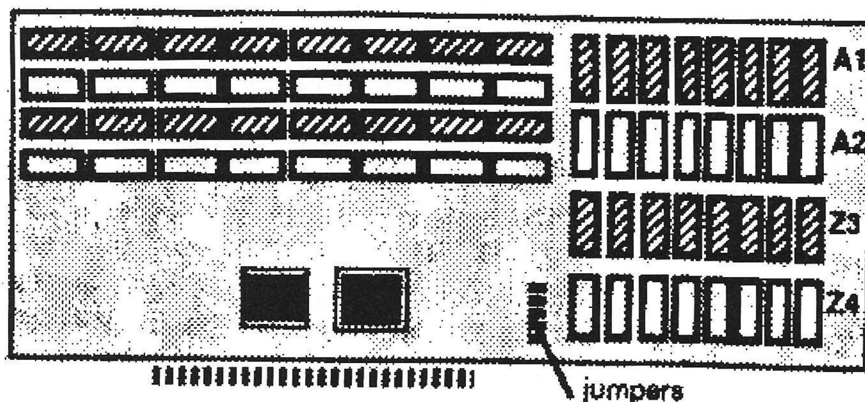
Meer RAM geeft meer Amiga-kracht. Dat behoeft beslist geen dure investering te worden, want met twee MegaByte extra komt de Amiga-gebruiker al een heel eind. Voor echt zware toepassingen komt 4 tot 8 MB extra in aanmerking.
De hier geteste 8-UP! (DIP)-kaart biedt een RAM-configuratie van 2 tot en met 8 MB aan standaard DIP-RAM. Voor de Amiga ziet deze kaart er uit als twee gescheiden PICs, d.w.z. wordt gezien als twee aparte uitbreidingskaarten. Elk van die kaarten kan 2 of 4 MB aan extra RAM-geheugen bevatten. U kunt de 8-UP!-kaart dus als 2, 4, 6 of 8 MB configureren.
Die 6-optie is zoals bekend belangrijk voor Amiga's met een Bridgeboard waar maximaal een 6 MB-configuratie mogelijk is.

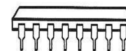
De kaart zelf

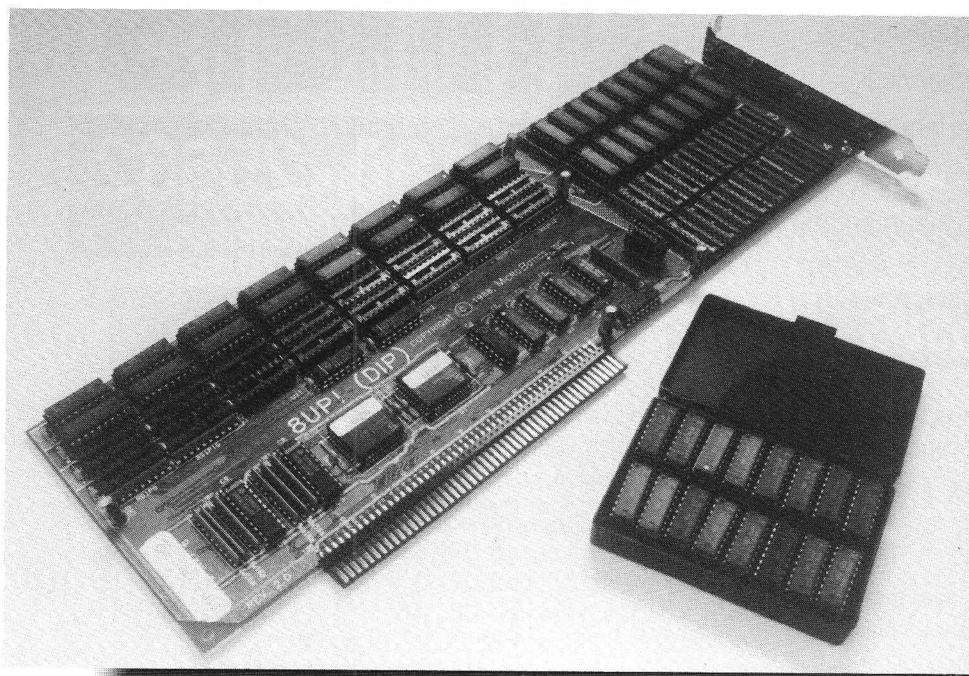
De MicroBotics 8-UP!-kaart bestaat uit een zero-wait state autoconfigurerende RAM-print met daarop 2 t/m 8 MB aan Dynamic Random Access Memory (DRAM)-chips.
Op deze kaart zitten de RAM- PICs "A" en "Z" (staat op de printplaat) verdeeld in vier groepen. Die groepen zijn A1, A2, Z3 en Z4. Elke groep bestaat op zichzelf weer uit 8 horizontale en 8 vertikale DRAM-chips.
MicroBotics beveelt aan om DRAM-chips met een minimale snelheid van 150 nanoseconde te gebruiken.
De 8-UP!-kaart is auto-installing en beschikt voor nadere configuratie-specificaties over 7 jumper-blocks.

De installatie

Na het verwijderen van de eenvoudige kartonnen verpakking en het roze anti-statische schuim houdt u een lange Amiga-uitbreidingskaart met 100- pins connec-







tor in uw handen. Leg deze kaart op een veilige plaats, liefst nog op het antistatische schuimplastic, neer. Schakel eerst de netstroom uit! Open nu de systeemkast door de vijf standaard Amiga-schroeven te verwijderen.
Een Amiga beschikt over vijf 100-pins slots voor uitbreidingskaarten voor de Amigabus. Bij voorkeur zet u de harddiskcontroller in het meest rechtse slot. De 8-UP! kan dan in elk ander slot gedrukt worden.
Dat drukken dient gelijkmatig en zonder forceren te geschieden. Natuurlijk dient u eerst een van de metalen vleugels te verwijderen en de kaart goed ten opzichte van de kaartgeleider en het connectorslot te positioneren.
Hanteer de kaart uitsluitend aan de randen en blijf met de vochtige, kleverige statische vingertjes van de elektronica af! Zit de kaart goed in de connector en geleiderail draai dan het vleugelschroefje aan. Komt er nu een ingewikkelde installatie-procedure? Het antwoord luidt ronduit nee. Bij onze 2 MB-uitvoering bleek de 8-UP! zichzelf prima te configureren. Het enige verschil was dat het booten iets langer duurde.

Bij het gebruik van turbocards dient u te bedenken dat de 8-UP! een 16-bits geheugenbus gebruikt. De 32-bits processor van de turbokaart kan het extra RAM dan alleen op 50% van de theoretische snelheid benutten.
Moet u nog een turbokaart aanschaffen dan kan een model met piggyback-geheugen- uitbreiding de voorkeur verdienen boven een losse 16-bits RAM-kaart.

Wel bijregelen

Importeur Amigis garandeert gebruiksklare levering en een volledige test van de 8-UP!. Ondanks dit kunnen er in de praktijk toch een aantal problemen ontstaan. Bijvoorbeeld bij oude spelletjes die niet met een geheugenuitbreiding willen werken. In een dergelijk geval kunt u het utility NoFastMem gebruiken.
Inherent aan de XT- en AT-bridgeboards is dat zij alleen in machines met maximaal 6 MB aan RAM-uitbreiding willen werken. Bij 8 MB weigeren deze kaarten gewoon dienst.
Het autoconfigureren werkt vanaf AmigaDOS 1.2. Bij oudere versies kunnen problemen optreden. In dat geval kunt u de autoconfiguratiejumpers 1 (PIC "A") en (2 PIC "B") verwijderen.
Ook bij het uitbreiden van de standaard 2 MB zult u jumpers moeten verstellen. De Nederlandse gebruiksaanwijzing legt duidelijk uit hoe dat moet en welke DRAM-chips u daarvoor kunt gebruiken.

Testsoftware

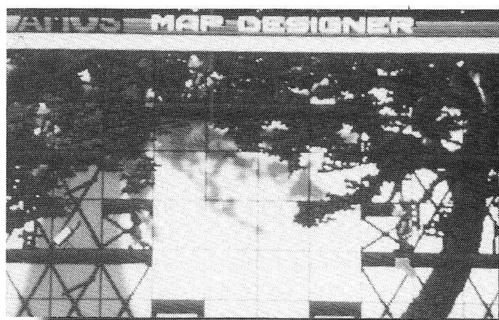
DRAM-chips hebben niet het eeuwige leven en ook bij aankoop van nieuwe chips kunnen er kneusjes tussen zitten. MicroBotics levert daarom een testdiskette bij. Met deze schijf in de DF0" kunt u via het zelflerende testproces de dader van een geheugenstoring gemakkelijk achterhalen.
De MicroBotics 8-UP! (DIP) is ons prima bevallen. Met name het installatiegemak en het probleemloos lopen op de Amiga 2000 sprak ons aan. Geen gekke koop voor *f* 798,- (2MB- uitvoering). Inlichtingen: Amigis, tel.: 01180-25632.
U.S.



AMOS

Heel de wereld uit z'n bol

Sinds een paar maanden is het zover: AMOS is uit. De langverwachte programmeertaal voor de Amiga loopt als een trein. Veel Atari ST-fans konden al genieten van een dergelijke Basic voor hun computer en nu eindelijk is het ook tijd voor de Amiga-gebruikers. Maar het was het wachten waard. De meesten verwachtten dat het een kopie zou zijn van STOS op de Atari, maar dat is niet het geval. Het programma is compleet herschreven en haast onherkenbaar gemaakt voor hen die STOS al kenden. Maar goed, laten we eens kijken wat we krijgen voor de *f* 159,- (een spotprijs overigens).



Om te beginnen zit het pakket in een stevige doos met programma- en data-diskette. Daarnaast een handleiding van zo'n 300 pagina's, een 'newsletter', een registratiekaart, een losse snelle index en een kaart voor het eerste gebruik met AMOS. Die laatste moet zeker worden gelezen, daar er wat veranderingen ten opzichte van het handboek zijn opgetreden.

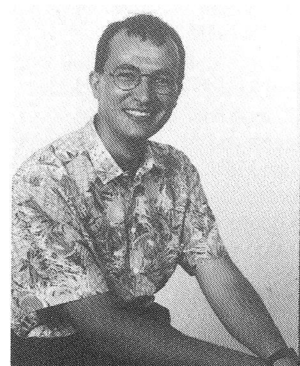
Programma

Een review maken over een programma dat meer dan 500 commando's kent, kan nooit compleet zijn in een aantal pagina's. Daarom kunnen we ze niet allemaal behandelen.
Amiga Basic kent iedere gebruiker. Je krijgt het immers als je een Amiga koopt. Iemand die eens wat wilde programmeren kon dat zonder extra uitgaven doen. Maar aan Amiga Basic kleven best wat nadelen, al was het maar de beperkte hoeveelheid commando's. Voor hen die al lang iets meer en vooral makkelijker wilden programmeren kunnen hun hart ophalen.

Zoals Francois Loinet, de maker van Amos, in ons vorige nummer al opmerkte werkt Amos, niet via de WorkBench. Het is wel mogelijk, maar wie Amos van de originele diskette opstart, zal merken dat er geen WorkBench bij aan te pas komt. Dit is gedaan om zoveel mogelijk geheugen beschikbaar te hebben voor het te maken programma. Amos kent namelijk geen werkgeheugen in die zin dat het maximaal de grootte heeft die de gebruiker aan RAM beschikbaar heeft. Iemand die geen geheugenuitbreiding heeft, kan Amos prima gebruiken.

Editor

Amos Basic werkt met een snelle editor, vooral als we de



vergelijking met Amiga Basic maken. De schuifregelaar doet een beetje denken aan STOS, een Amos-ogende programmeertaal voor de Atari ST. In feite doet de editor meer aan een tekst-editor denken, met name wat betreft de 'zoek- en vervangfunctie', 'bewaren in ASCII' en 'meerdere programma's tegelijkertijd in het geheugen laden'. Zoals gezegd is het maximale geheugen de capaciteit van het in te laden programma.
Amos kent ook een directmode. Een druk op Esc en er springt een tweede kleiner venster op de voorgrond waarin een groot aantal commando's direct kunnen worden uitgevoerd. De functietoetsen (met en zonder shift) kunnen worden geprogrammeerd.
Voor Amiga-fans die al langer met standaard Amiga-Basic werken een geruststelling: Amos kent alle commando's van Amiga-Basic.
Nummering van de regels is binnen Amos niet meer nodig.
Wel anders is de opbouw van de programma's die met on-



derprogramma's werkt. Deze manier van werken doet wat denken aan GFA-Basic. Het is te vergelijken met de goto-functie maar bij deze verwijzingen staan complete routines. Een specialiteit van Amos is dat het met een enkel commando hele array's kan doorzoeken, het daarna een requester laat zien, waarin alle files uit de actuele directory te zien zijn. Amos maakt daarbij geen gebruik van de normale Amiga-wildcards '#' en '?', maar van '*' van MS-DOS.
Met fonts gaat Amos ook erg gemakkelijk om. Alle lettertype worden in een regel ingelezen en daarna met een commando gezet. Ook teksten cursiveren of onderstrepen gaat met een simpel commando ON en commando OFF.
Bij ieder commando blijkt weer hoe gemakkelijk Amos is opgezet. Zelfs iemand die nog nooit geprogrammeerd heeft, hoeft geen enkele kennis van de opbouw van het Amiga-DOS te hebben. Logische woorden (in het Engels, dat wel) zoals Hires en Lores zeggen de gebruiker genoeg en Amos ook. Voor wat de resolutie van het scherm betreft, accepteert Amos Basic ook Hold and Modify en Dual Playfield.
Wie denkt dat Amos, omdat het zonder pull-down menu's werkt, ze niet kan maken, heeft het helemaal mis. Het is zelfs mogelijk om bijna oneindig submenu's te maken. Codes als INPUT$ worden gewoon geaccepteerd, maar Amos gaat nog een stapje verder. Hij herkent zelfs het indrukken van Shift, Control of Alt. De cursorvorm is ook helemaal aan te passen.

Grafisch

Voor de fans van IFF-plaatjes en alles wat met grafische beelden te maken heeft, is er goed nieuws: Amos barst van de grafische mogelijkheden. Zo dook al weer even geleden een systeemcommando op waarmee een regenboog-effect op het scherm kon worden verkregen. Amos heeft die mogelijkheid ook maar dan met een commando: RAINBOW. IFF wordt ondersteund, zelfs zo dat plaatjes kunnen worden opgeroepen. Dus eerste een basisontwerp maken waarover in een later stadium de sprites bewegen, is dus mogelijk.

AMAL en BOB

Om ze te laten bewegen moet heel simpel een begin-, een eindpunt en de snelheid worden ingegeven en de boel loopt. Voor botsingen tussen sprites kunnen geluiden of meldingen worden gedefinieerd.
Voor sprites en BOB's heeft de programmeur AMAL, Amiga Animation Language, ontwikkeld. Dit is in feite een programma op de achtergrond, dat onafhankelijk van Amos loopt. Is dat eenmaal het geval, dan kan AMAL 16 sprites of BOB's tegelijkertijd sturen.
Qua muziek zitten we bij Amos ook goed. Explosie, schot en klok zitten al standaard voorgeprogrammeerd,

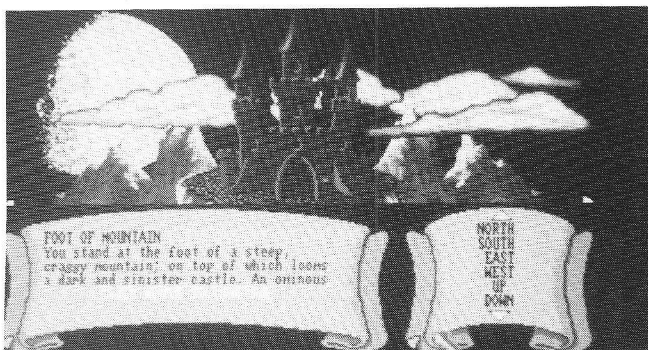
```
'
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'        Castle AMOS
'
'   By A.M.Kalim & P.J.Hickman
'
'      1990 Mandarin Software
' ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Set Buffer 12
' ----- Free memory -----
Close Editor
If Chip Free250000
  ENOUGH_MEM=True
Else
  ENOUGH_MEM=False
End If
Limit Mouse 128,167 To 447,235
Dir$="Amos_Data:"
Auto View Off
Curs Off
Flash Off
Colour 0,$0
Colour 1,$0
Rem -----------------------------------
Rem ----- Data declaration Segment ------
Rem ---------------------------------
LOCATION_SCREEN=0
COMMAND_SCREEN=1
COMMANDS=13
OBJECTS=26
LOCATIONS=5
MESSAGES=12
Dim COMMAND$(COMMANDS),LIST(6)
Dim MAP(LOCATIONS,4),MAP$(LOCATIONS)
Dim OBJECT(OBJECTS),OBJECT$(OBJECTS,2)
Dim MESSAGE$(MESSAGES)
Rem -----------------------------------
SETUP_ENVIRONMENT
INITIALISE_COMMANDS
INITIALISE_MAP
INITIALISE_OBJECTS
INITIALISE_MESSAGES
'
```

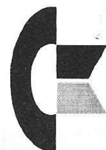
Het begin van Castle AMOS, meegeleverde software bij Amos

maar het is ook mogelijk Amos hele muziekstukken te laten spelen. Wie overtuigd wil raken van Amos voordat hij zelf begint, hoeft alleen maar even een meegeleverd programma te runnen en ziet wat we bedoelen.

Snelheid

Opvallend is dan ook de snelheid waarmee het programma werkt, helemaal in vergelijking met Amiga-Basic. Wat de snelheid betreft wachten we met spanning op de compiler die eraan zit te komen.
Zij die Amos serieus willen gebruiken, moeten niet terugschrikken van wat Engelse taal. Het programma werkt natuurlijk in het Engels, maar ook het handboek is helemaal in die taal geschreven. Het handboek is weliswaar handzaam, maar ingebonden. Het resultaat hiervan is, dat het boek nooit bij een pagina blijft liggen. Een ringband was weliswaar beter geweest.

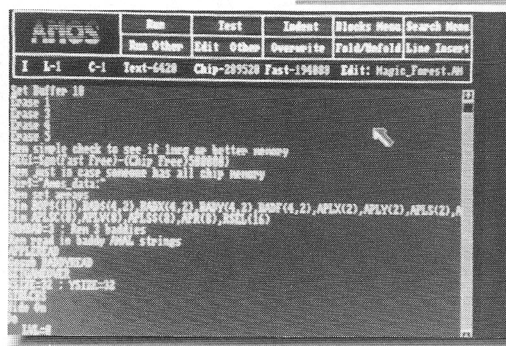




Met een aantal meegelevere hulpprogramma's, die onder Amos draaien, kunnen nog een aantal zaken worden geregeld, zoals keyboard-layout. Maar daarvoor ontbreken de gegevens in het handboek. Wel handig is de zogenoemde Handy index, een 12-pagina's tellend boekje met een index en alle foutmeldingen. De index staat overigens niet in het handboek zelf.

Conclusie

Amos is goed gelukt. Wie het vergelijkt met Amiga-basic weet niet wat-ie ziet. Maar daarvoor betaal je dan ook f 159,-. Heb je geen verstand van programmeren en al helemaal niet van het basissysteem van de Amiga, dan is Amos perfect geschikt. Maar ook ervaren programmeurs kunnen veel plezier aan Amos beleven, helemaal als straks de compiler uit is.




```
Procedure CHECK_MOUSE
  Shared COMMAND$(),LIST()
  Screen 1
  Window 2
  DISPLAY_LIST
  Repeat
    OLDZONE=0
    EXECUTE_COMMAND=False
    Repeat
      NEWZONE=Hzone(1,X Mouse,Y Mouse)
      If NEWZONE
        If NEWZONEOLDZONE
          If OLDZONE0
            Centre At(,OLDZONE-1)+COMMAND$(LIST(OLDZONE))
          End If
          Inverse On
          If NEWZONE0
            Centre At(,NEWZONE-1)+COMMAND$(LIST(NEWZONE))
          End If
          Inverse Off
          OLDZONE=NEWZONE
        End If
      End If
    Until Mouse Click
    If NEWZONE0 and NEWZONE
      EXECUTE_COMMAND=True
    End If
    If NEWZONE=7 Then REDRAW_LIST[1] : DISPLAY_LIST
    If NEWZONE=8 Then REDRAW_LIST[2] : DISPLAY_LIST
  Until EXECUTE_COMMAND=True
  Rem ------------------------------
  Rem ------ Process Commands ------
  Rem ------------------------------
  If LIST(NEWZONE)=1 and LIST(NEWZONE)6
    COMMAND_MOVE[LIST(NEWZONE)]
  End If
  If LIST(NEWZONE)=7 Then COMMAND_PICTURE : Pop Proc
  If LIST(NEWZONE)=8 Then COMMAND_LOOK : Pop Proc
  If LIST(NEWZONE)=9 Then COMMAND_EXAMINE : Pop Proc
  If LIST(NEWZONE)=10 Then COMMAND_PRESS : Pop Proc
  If LIST(NEWZONE)=11 Then COMMAND_READ : Pop Proc
  If LIST(NEWZONE)=12 Then COMMAND_INVENTORY : Pop Proc
  If LIST(NEWZONE)=13 Then COMMAND_QUIT : Pop Proc
End Proc
```

Een voorbeeld van een procedure, zoals die in AMOS is gedefinieerd

SPELLEN

Sierra komt met Leisure Suit Larry 3 en veel meer

Sierra is voor vele mensen een begrip geworden. Zoals de kleuters het plankje leren met aap, noot, mies, zo leert vrijwel elke gebruiker van PC en Homecomputer zijn systeem onder de knie te krijgen met Larry, King's Quest en Space Quest. Maar voor hen die dit type spel nog niet kennen, eerst een uitleg van de Sierramethode.

De spellen zijn nagenoeg allemaal hetzelfde qua opbouw, je bestuurt de hoofdrolspeler met behulp van cursortoetsen, joystick of muis door de vele schermen vol gevaren, nuttige objecten en uitdagingen. Sierra spellen zijn echter geen actiespellen. Het zijn voornamelijk adventures. Elk is gebaseerd op een hoofdgegeven. Het opvallende is echter dat elke groep meerdere spellen heeft. Met tussenpozen van ongeveer één jaar volgden zij elkaar op en de verzameling blijft groeien.
Een tweede feit is dat de gebruiker ze blijft kopen. Sierra is, zo kan men veilig stellen, met een winnende formule gekomen. Verder doet Sierra veel moeite om de gebruiker van een spel te laten genieten. Het spel wordt veelal geleverd met een uiteenlopende verzameling van drivers zodat de meest vreemde beeld- en geluid-combinaties mogelijk zijn.



Handleiding

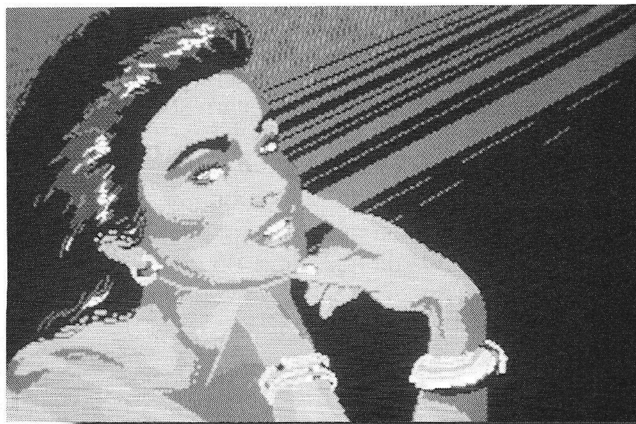
Ook de handleiding getuigt van afwerking. Geen saai boekje dat gaat over het installeren en laden van een spel, maar een fraai boekwerk met deze informatie, alsmede een achtergrondschrift met bijkomende informatie. Of het nu een perkament is met het gebod van een koning of een tactische manual met een kleurenkaart van de wereld, men kan er zeker van zijn dat men iets krijgt dat een lust voor het oog is.
Het beeld is zeer gedetaillieerd en laat zich simpel manipuleren. De hoofdrolspeler kan ook makkelijk worden bestuurd. Als men hem namelijk over een brug beweegt, moet men niet over de rand lopen. Hij blijft dan niet door de lucht lopen, maar zal een vroegtijdige dood vallen. Deze spellen zitten vol van dit soort geintjes en kunnen dus niet beschouwd worden als een saai adventure, maar als een vrolijk en fleurig geheel.

Nieuw

Sierra is onlangs voor de Amiga met een nieuwe collectie spellen uitgekomen waarvan enige opvolgers maar ook een verzameling nieuwe spellen. Het opvallende aan deze spellen is het feit dat men het maximum aan kwaliteit uit deze spellen heeft willen halen. Dat is ze ook gelukt, de gebruiker moet er echter wel rekening mee houden dat een geheugenuitbreiding van 512 kB noodzakelijk is om de spellen te kunnen spelen.
Sierra is ook een van de weinige bedrijven die blijft volharden dat spelbeveiligingen beter in de handleiding kunnen zitten dan in de diskette (als het tenminste geen Nederlands paspoort betreft), dus installeren op harddisk is dan ook geen enkel bezwaar.

Leisure Suit Larry 3

De eerste in zijn cyclus en zonder meer de mooiste. Hij gaat verder waar deel 2 stopte. Voor die gebruikers die de andere delen nooit gespeeld hebben, geen zorg. Elk spel kan onafhankelijk van elkaar gespeeld worden. In deze categorie staat de jacht op liefde en sex centraal. Het spel begint op een uitkijkpost van een tropisch eiland. Larry geniet van het uitzicht op zee en van het zicht op een uitkledende overbuurvrouw. Uitgegluurd gaat hij naar huis om erachter te komen dat zijn



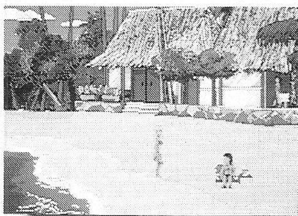




vrouw zich bezig houdt men het bedriegen van haar man. Larry wordt geconfronteerd met het verlies van vrouw, huis en werk en heeft hiervoor teruggekregen zijn status als vrijgezel. Het is jouw taak geworden ervoor te zorgen dat zoveel mogelijk vrouwen van jou kunnen genieten en jij van hen.
Een leeftijdstest aan het begin van het spel bepaalt wat de speler te zien krijgt. Bijvoorbeeld de luxaflex van een huis waar een dame zich uitkleedt gaan niet omlaag als men meerderjarig is. Ook een leuk gegeven is het rollenpatroon in het spel. Het eerste gedeelte speelt men de rol van Larry gewapend met pulsating pectorals, zo bekleed je de tweede helft van het spel de rol van Passionate Patti, die verliefd is geworden op Larry en hem probeert te bevrijden uit handen van kannibalistische amazone's. Humor, sex en een afwezig blijven van wetenschap maken dit spel tot een waardevolle aanschaf.

Iceman

Als commandant bij de marine word je gedurende je opwindende vakantie op Tahiti gestoord met het bericht dat je je direct moet melden voor militaire dienst. Eén telefoontje en het is zover. Pak je tickets voor Washington en meld je bij jouw commanderend generaal in het Pentagon voor dienst. Daarna krijg je als opdracht je te melden op Hawaï bij één van de meest geavanceerde onderzeeërs die de marine kent. De U.S.S. Blackhawk. Jouw taak? Bevrijd de Amerikaanse ambassadeur.
Hij is gekidnapped door Libische terroristen. Om dit doel te bereiken zul je de U.S.S. Blackhawk door diverse gevaren moeten loodsen, variërend van Russische oorlogsbodems tot Titanic-liefhebbende ijsbergen. Opvallend hierbij is het feit dat men dit zoveel mogelijk via actieonderdelen heeft laten verlopen. Variërend van dobbelen tot het zinken van een Russische kruiser en een Russische Alpha-sub, het manoeuvreren door een gebied met ijsbergen. Een soort van kleine simulator in een adventure.
Het volgende onderwerp is duiken (tjonge jonge, water is volop aanwezig in het spel). Ook ontmoetingen met vrouwelijke spionnen staan op het spel. 003 is volop in actie. Al met al is het een van de meer technisch beheb de adventures van Sierra.



Space Quest 3

Deze is duidelijk gebaseert op het onderwerp sciencefiction. Je speelt hier de rol van Roger Wilco. Het spel gaat door waar deel 2 eindigde. Roger ligt in een noods-Shuttle en is op weg naar de beschaving. Bij de intro wordt duidelijk dat hij door een junk carrier ingenomen wordt. Als hij in een opslagruimte arriveert, wordt hij gewekt. Het wordt duidelijk dat, als hij een kans wil hebben, het noodzakelijk is uit dit schip weg te komen.
Voor zover de handleiding verder uitlegt zijn de twee beroemde software ontwerpers van Andromeda ontvoerd door software-piraten. Ongetwijfeld zal dit een taak worden voor onze Space Roger.
De beelden zien er goed uit en er zitten leuke komische momenten in, zoals bijvoorbeeld de scene waarbij Roger overvallen wordt door de plaatselijke rattenbrigade. Deze actie wordt voorafgegaan door de muziek van Jaws.

Conquests of Camelot

Het is de tijd van koning Arthur en de ridders van de ronde tafel. Als Arthur reis je door Engeland en het Heilige Land op zoek naar de Heilige Graal. Dit is behoorlijk opwindend. Als je er in slaagt om de graal te vinden en mee te nemen, betekent dit dat de graal in Engeland ligt. Zodoende heeft Steven Spielberg $ 50,000,000 dollar voor niets uitgegeven om hem in de Vallei van de Halve Maan te gaan zoeken. Maar terug naar dit spel.
Een van de belangrijke aspecten is het feit dat goden en godinnen (die van de liefde) je met raadsels zal confronteren die je met je gezonde verstand zult moeten oplossen. Een tweede is het lansgevecht tegen de zwarte ridder. (Waarom is de zwarte ridder nou altijd de slechterik?) Aangezien dit in die tijd belangrijk was zul je het ook nu moeten beheersen. Gelukkig bleek het niet al te moeilijk te zijn.



The Colonel's Bequest

Het eerste wie-heeft-het-gedaan mysterie van Sierra. Dit is een van de weinige spellen waarbij een vrouw de spil van het spel is. Het begint met het feit dat de kluizenaar Kolonel Dyon een reünie organiseert in de diepe Bayous van Louisiana. Het is 1925 en de wilde jaren '20 zijn op zijn hoogst. Je bent getuige van het feit dat de kolonel zijn miljoenen nalaat aan alle nazaten die aanwezig zijn. Met de clausule dat als voor de dood van de kolonel een nazaat sterft, de rest zijn deel zal krijgen. En dat moet fout gaan.
Terwijl woede en verraad aan de orde van de dag zijn, moet je de moordenaar vinden voordat nog een slachtoffer op de groeiende stapel van lijken komt.
De beste methode om aan tips te komen is om alles grondig te doorzoeken en als Sherlock Holmes te werk te gaan. Het meeste kom je over een persoon te weten, als je er niet bent en hij er wel is.
Voor het stillen van honger moet je het moeras bezoeken. Al met al is dit een leuk spel met een goed einde. De schuldige is niet degene die je denkt dat het is. Maar dat is het dan ook nooit.

King's Quest IV

Het spel waarmee het begon. Wederom is men erin geslaagd om grenzen te verleg-





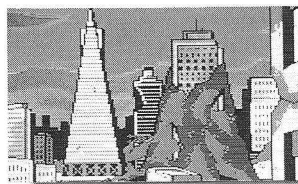
gen in de adventure-wereld. In vergelijking tot zijn voorgangers is KQIV een juweel van een programma. Alle grenzen op het gebied van beeld en geluid zijn doorbroken met dit programma.
De stunts lopen uit een van het kabbelende water tot de Pauw die bij benadering een mooie verentooi ten toon spreidt.
De oude koning Graham is op leeftijd gekomen en neemt het besluit zijn avonturenhoed af te geven aan zijn opvolger. Er zijn twee gegadigden: een zoon en een dochter. Hij gooit zijn hoed neer voor de twee, zodat de besluitvaardige hem op kan nemen voor de tijd die komen gaat.
Plotseling gaat er iets mis. De koning wordt door ziekte getroffen en ligt stervend op bed. Terwijl de familie het ziekenbed bewaakt loopt dochter Rosella naar de troon en kijkt naar de hoed. De magische spiegel komt tot leven en vertelt van het magische fruit dat de koning redden kan. Er is echter weinig tijd. Rosella aarzelt geen moment en springt in de spiegel op weg naar de redding.
Deze actie brengt haar naar het land Tamir, een land vol legendarische wezens. Vele gevaren zal zij moeten overwinnen, vele beproevingen moet zij doorstaan om het fruit te bemachtigen. Wat vooral leuk is, is het feit dat men veelal gebruik maakt van bekende sprookjes als achtergrond voor het geheel. Cupido, de walvis en diverse andere wezens zullen menig dag en nacht de Amiga verblijden. Voor de toeristische route: bezoek het kerkhof 's nachts.

Hero's Quest

Zo, dus je wou de held uit gaan hangen? Bereid je dan voor op een van Sierra's meest uitdagende adventures die in zijn collectie te vinden is. Hero's Quest (HQ) verschilt aanzienlijk ten opzichte van zijn broertjes. Alhoewel men hetzelfde vertrouwde beeld tegemoet gaat, is dit niet van toepassing met zijn spelvorm. Hier begint men in feite met een basis-karakter, vechter, magiër of dief. Dit basis-karakter zal men moeten scherpen door kennis wil men de tovenaar Baba Yaga verslaan. Deze methode om je karakter te ontwikkelen houdt dan ook in, dat hetzelfde probleem verschillende methode's kent om overwonnen te worden. Bijvoorbeeld de vechter zal een zware poort openen met spierkracht, de magiër heeft echter een spreuk nodig om hetzelfde te bereiken. Het scherpen van kennis houdt dus ook in dat leermeesters op wapen- en magie-gebied gevonden moeten worden om verder te kunnen. Het geheel ziet er goed uit en is het meest te vergelijken met zijn King's Quest-broertjes.

Manhunter San Francisco

Manhunter is de uitzondering op alle adventure's van Sierra. Als eerste beweeg je het figuur niet op de zelfde wijze. Als Manhunter ben je in dienst van de Orbs. Dit is een levensvorm die wegens hun technische vooruitgang de aarde hebben veroverd. Orbs zijn gigantisch zwevende ogen. Zij hebben willekeurig mensen aangewezen om hun te dienen in de onderdrukking. Jouw taak: spoor misdadigers op en rapporteer aan de sectie Orb. In dit spel beweeg je een cursor die op verschillende lokaties een andere vorm aanneemt en daardoor een handeling aanduidt. Als je de cursor op een lichaam of object duid zal de cursor veranderen in



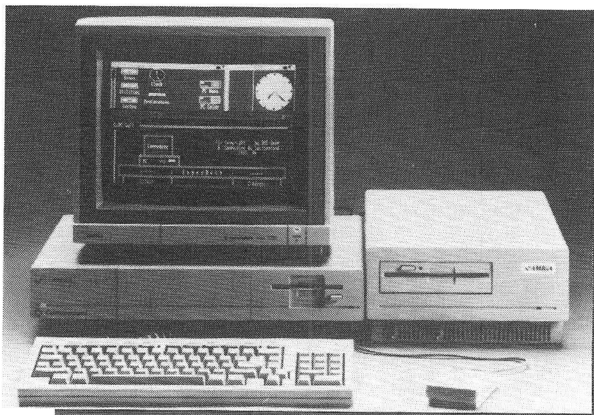
een hand of bijvoorbeeld een vergrootglas. De vorm die hij aanneemt is niet te beïnvloeden.
Het bereiken van diverse lokaties van een reiscomputer. Hierin zit een deel van het spel. De Orbs stellen dat je enkel lokaties kunt bezoeken als zij een deel van het onderzoek zijn. Een adres krijg je door via een tracker mensen te volgen en die lokaties te bezoeken. Ook komen lokaties beschikbaar als je mensen identificeert. De computer zal dan hun adres geven, die dan ook bezocht kan worden. Bewegingen bereik je met het aanduiden van eindbestemmingen op de reiscomputer.
De Manhunter-serie is een van de meest bizarre in de verzameling. Dit deel speelt zich af in de donkere toekomst van 2004, San Francisco is dit keer de achtergrond voor een bizar Plot dat de Orbs complete dominantie moet geven. Ze worden bijgestaan door moordende robot's en menselijke verraders. Men heeft in dit spel leuk gespeelt met de beelden die San Francisco onderscheiden van andere steden, zoals Fisherman's Wharf, Coit Tower, Chinatown, Alcatraz en vele andere lokaties. De Manhunter-collectie kent volgens zeggen de minste volgelingen van allen.
Er was hier ook gekozen voor een geheel ander uiterlijk. Het is echter een spel dat qua uitdaging niet onderdoet voor zijn voorganger. Als men wil genieten van Bizarre Science Fiction en wil bewijzen dat de rol van Manhunter ook tegen de onderdrukkers gebruikt kan worden krijgt men hierin een goede kans. Grafisch is hij niet zo goed als je de kwaliteit van Larry of King's Quest voor ogen houdt, maar evenzo een goede en aparte aanwinst op de Sierra-plank.





Railroad Tycoon van Microprose

Het programma waarmee Microprose de simulatormarkt probeert te veroveren is Railroad Tycoon. Hiermee doen ze dan eindelijk een intrede met een wat luchtere simulator die getuigt van originaliteit. Het is jouw taak om een spoorlijn op te zetten in één van de vier mogelijke terreinen. Het spel heeft de mogelijkheden West-Amerika, Oost-Amerika, Engeland en Europa.



Je moet er niet vanuit gaan dat je een spoorlijn neerzet tussen Los Angeles en San Francisco en daarmee word je rijk, want dat is niet zo. Het spel gaat met name om de vraag over het leveren van eindprodukten aan de stad.
Om een voorbeeld te geven: een haven levert hop. Deze moet vervoerd worden naar een brouwerij, die maakt er bier van hetgeen weer naar steden vervoerd moet worden voor consumptie. Er heerst dus vraag en aanbod. Rijk word je er aan als je dit met een minimum aan kosten efficiënt aan de man kunt brengen.
Er zijn nog meer factoren zoals de aanleg van een spoorlijn. Als je de lijn aanlegt, moet je goed in de gaten houden dat je in bewoonde gebieden plaatsingsrechten moet betalen. Als je deze door twee steden plaatst, ben je al snel een fortuin kwijt. Ook het updaten van de spoorlijn moet goed in de gaten gehouden worden.
Als een station met de tijd belangrijker wordt, dan begint de aanwezigheid van opslagdepots, onderhoudscentra en wisselstations een must voor de ambitieuze investeerder te worden.
Ook de ontwikkeling staat niet stil. Gedurende het spel wordt bij de speler het nieuwste op het gebied van locomotieven aangeboden. Het bijhouden van deze elementen is een must, want tijd is geld en er is natuurlijk nog altijd de competitie.
Het grafische gedeelte van het spel ziet er goed uit. Behalve dat op het gedetailleerde gebied elk huis identiek is, wat snel gaat vervelen. Met het plannen van treinen is dat niet het geval en dat is dus aan te bevelen.
Het goed runnen van een spoorlijn vereist het gebruik van de grijze cellen en is een goede uitdaging. Als je het spel dieper beheerst kun je over gaan op moeilijker opties, zoals gebruik maken van zwaardere tegenstanders en het gebruik van economische blokkades. Op de zwaarste stand gaat zelfs de directeur van de NS een eerste klas nachtmerrie tegemoet.
Dit programma is op het ogenblik alleen verkrijgbaar voor de PC (CGA/EGA/VGA). Begin volgend jaar zal dit programma voor de Amiga beschikbaar zijn.

NIEUW! - NIEUW! - NIEUW! - NIEUW! - NIEUW!

Eindelijk uit: dé Nederlandse

GEOS 2.0-basishandleiding

Dit boek is speciaal geschreven voor hen die werken met een Commodore 64/128 en het geavanceerde besturingssysteem GEOS 2.0.
Het eerste Nederlandse boek voor de rechtgeaarde GEOS-gebruiker, die de mogelijkheden van dit pakket ten volle wil benutten.

Het boek beschrijft naast de werking van GEOS 2.0, uitvoerig het gebruik van de Desktop, GeoPaint, GeoWrite en alle desk-accessoires. Tevens komen applicaties als GeoSpell en GeoMerge aan de orde. Verder wordt in dit boek aandacht besteed aan de andere GEOS-applicaties zoals GeoCalc, GeoFile, GeoPublish en GeoChart.
Indien u GEOS op de manier gäat gebruiken waarvoor het bedoeld is, zult u dit boek op zijn juiste waarde weten te schatten.

f 34,95

**U kunt de GEOS 2.0 - basishandleiding bestellen door overmaking van het betreffende bedrag (incl. verzendkosten) op gironummer 5641219 t.n.v. SALASAN Amsterdam o.v.v. 'GEOS basishandleiding'.
Dealer aanvragen welkom.**



LPG-BEREKENING

Het door ons ontwikkelde programma is zoals de kop al doet vermoeden een klein rekenprogramma. Het zal u vermoedelijk inmiddels wel zijn opgevallen dat de brandstofprijzen al met verschillende procenten zijn gestegen. Voor de persoon met een LPG tank in de auto valt het gelukkig allemaal wel mee. Maar wanneer je die, nog, niet bezit is het wel eens leuk om te weten of het voor u ook goedkoper kan, als benzine vergasser. Naast de aanschaf en verhoogde wegenbelasting is er echter een klein nadeel. Uw auto verliest ongeveer 10 pk aan vermogen op LPG. Ook zal het verbruik met 20% toenemen. Dit is in ons programma verwerkt. Maar u zal heel wat voordeliger kunnen rijden.

De werking

Als eerste zal het programma u om enige invoer vragen. Zoals de aanschafprijs van de LPG installatie, het aantal termijnen waarin u de installatie wilt gaan afschrijven, de wegenbelasting voor benzine en voor gas enz. enz. Bij het postkantoor kunt u de tabel, kosteloos, ophalen waarin de bedragen van LPG, DIESEL, en 'normale' brandstof zijn opgenomen. Deze zult u nodig hebben tijdens het runnen van ons programma. De rest wijst zich vanzelf.

Type computer

Dit programma zal met enige aanpassingen op elk type computer kunnen draaien. Maar u kunt rustig zonder enige aanpassingen dit programma draaien op de C64, C128 en de Amiga. Natuurlijk ook op andere computers maar dat is nu niet aan de orde. Voor elk van de drie hierboven genoemde computer is een aparte listing in dit blad opgenomen.

Tot slot

Veel plezier met dit door ons ontwikkelde programma. Ook nu weer alle leuke op en aanmerkingen zijn van harte welkom.

Johan & Johan

Programma.... Amiga-versie

```
====================================
100 REM *******************************
110 REM *  LPG BEREKENING PROGRAMMA   *
120 REM *        AMIGA VERSIE         *
130 REM *    DOOR JOHAN & JOHAN       *
140 REM *******************************
150 REM
160 PRINT CHR$(12);
170 PRINT  "--------------------------------"
180 PRINT " LPG BEREKENING PROGRAMMA,
    VERSIE 1.00"
190 PRINT "--------------------------------"
200 PRINT:INPUT "Prijs van de LPG
    installatie:";ap
210 IF ap=0 THEN 200
220 PRINT:INPUT "Aantal termijnen voor
    afschrijving:";at
230 IF at=0 THEN 220
240 PRINT:INPUT "Wegenbelasting voor
    benzine:";wb
250 IF wb=0 THEN 240
260 PRINT:INPUT "Wegenbelasting voor
    LPG:";wg
270 IF wg=0 THEN 260
280 PRINT:INPUT "Inhoud benzinetank:";bt
290 IF bt=0 THEN 280
300 PRINT:INPUT "Benzine prijs:";bp
310 IF bp=0 THEN 300
320 PRINT:INPUT "LPG prijs:";gp
330 IF gp=0 THEN 320
340 PRINT CHR$(12);
350 PRINT "--------------------------------"
360 PRINT "    Kosten en afschrijving per
    maand"
370 PRINT "--------------------------------"
380 PRINT
400 PRINT USING"Afschrijving    F
    #####.##";(ap/at)
420 PRINT USING"Wegenbelasting F
    #####.##";(wg/12)
430 PRINT "                 ========== +"
440 PRINT USING"Totaal lasten   F
    #####.##";(ap/at)+(wg/12)
450 PRINT:PRINT
460 PRINT USING"In totaal dus
    F#####.##";(ap/at)+(wg/12)-(wb/12);
465 PRINT " extra p/mnd"
470 PRINT
490 PRINT USING"Tank benzine    F
    #####.##";(bt*bp)
510 PRINT USING"Tank LPG        F
    #####.##";(bt*1.2)
520 PRINT "                 ========== -"
530 PRINT USING"Per tank        F
    #####.##";(bt*bp)-(bt*1.2)
540 PRINT
550 PRINT "LPG kan voor u p/mnd voordeliger
    zijn"
560 hu=((ap/at)+ (wg/12))/((bt*bp)-((bt*1.2
    )*gp))
570 PRINT
580 PRINT USING"U moet nu dan ##### liter
    ";(hu*bt);
585 PRINT "p/mnd tanken"
590 PRINT
600 PRINT USING"Dit is voor F #####.##
    ";((hu*bt)*bp);
605 PRINT "aan benzine";
```



programma.... C128-versie

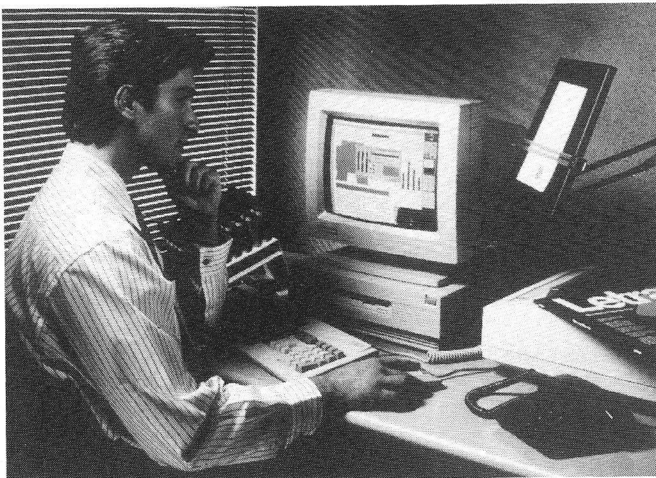
```
==================================
100 REM ******************************
110 REM *  LPG BEREKENING PROGRAMMA   *
120 REM *          C-128 VERSIE       *
130 REM *    DOOR JOHAN & JOHAN       *
140 REM ******************************
150 REM
160 PRINT CHR$(147);
170 PRINT "--------------------------------"
180 PRINT " LPG BEREKENING PROGRAMMA,
    VERSIE 1.00"
190 PRINT "--------------------------------"
200 PRINT:INPUT "Prijs van de LPG
    installatie:";ap
210 IF ap=0 THEN 200
220 PRINT:INPUT "Aantal termijnen voor
    afschrijving:";at
230 IF at=0 THEN 220
240 PRINT:INPUT "Wegenbelasting voor
    benzine:";wb
250 IF wb=0 THEN 240
260 PRINT:INPUT "Wegenbelasting voor
    LPG:";wg
270 IF wg=0 THEN 260
280 PRINT:INPUT "Inhoud benzinetank:";bt
290 IF bt=0 THEN 280
300 PRINT:INPUT "Benzine prijs:";bp
310 IF bp=0 THEN 300
320 PRINT:INPUT "LPG prijs:";gp
330 IF gp=0 THEN 320
340 PRINT CHR$(147);
350 PRINT "--------------------------------"
360 PRINT "    Kosten en afschrijving per
    maand"
370 PRINT "--------------------------------"
380 PRINT
400 PRINT USING"Afschrijving   F
    #####.##";(ap/at)
420 PRINT USING"Wegenbelasting F
    #####.##";(wg/12)
430 PRINT "                ========== +"
440 PRINT USING"Totaal lasten  F
    #####.##";(ap/at)+(wg/12)
450 PRINT:PRINT
460 PRINT USING"In totaal dus
    F#####.##";(ap/at)+(wg/12)-(wb/12);
465 PRINT " extra p/mnd"
470 PRINT
490 PRINT USING"Tank benzine   F
    #####.##";(bt*bp)
510 PRINT USING"Tank LPG       F
    #####.##";(bt*1.2)
520 PRINT "                ========== -"
530 PRINT USING"Per tank       F
    #####.##";(bt*bp)-(bt*1.2)
540 PRINT
550 PRINT "LPG kan voor u p/mnd voordeliger
    zijn"
560 hu=((ap/at)+(wg/12))/((bt*bp)-
    ((bt*1.2)*gp))
570 PRINT
580 PRINT USING"U moet nu dan ##### liter
    ";(hu*bt);
585 PRINT "p/mnd tanken"
590 PRINT
600 PRINT USING"Dit is voor F #####.##
    ";((hu*bt)*bp);
605 PRINT "aan benzine";
```

programma... C64-versie

```
==================================
100 REM ******************************
110 REM *  LPG BEREKENING PROGRAMMA   *
120 REM *          C-64 VERSIE        *
130 REM *    DOOR JOHAN & JOHAN       *
140 REM ******************************
150 REM
160 PRINT CHR$(147);
170 PRINT "--------------------------------"
180 PRINT " LPG BEREKENING PROGRAMMA,
    VERSIE 1.00"
190 PRINT "--------------------------------"
200 PRINT:INPUT "Prijs van de LPG
    installatie:";ap
210 IF ap=0 THEN 200
220 PRINT:INPUT "Aantal termijnen voor
    afschrijving:";at
230 IF at=0 THEN 220
240 PRINT:INPUT "Wegenbelasting voor
    benzine:";wb
250 IF wb=0 THEN 240
260 PRINT:INPUT "Wegenbelasting voor
    LPG:";wg
270 IF wg=0 THEN 260
280 PRINT:INPUT "Inhoud benzinetank:";bt
290 IF bt=0 THEN 280
300 PRINT:INPUT "Benzine prijs:";bp
310 IF bp=0 THEN 300
320 PRINT:INPUT "LPG prijs:";gp
330 IF gp=0 THEN 320
340 PRINT CHR$(147);
350 PRINT "--------------------------------"
360 PRINT "    Kosten en afschrijving per
    maand"
370 PRINT "--------------------------------"
380 PRINT
390 d1=INT(((ap/at)*100)+.5)/100
400 PRINT "afschrijving   F ";d1
410 d2=INT(((wg/12)*100)+.5)/100
420 PRINT "wegenbelasting F ";d2
430 PRINT "                ========== +"
440 PRINT "totaal lasten  F ";(d1+d2)
450 PRINT:PRINT
460 PRINT "in totaal dus
    F";(d1+d2)-INT((((wb/12))*100)+.5)/100;"ex
    tra p/mnd"
470 PRINT
480 d3=INT(((bt*bp)*100)+.5)/100
490 PRINT "tank benzine   F ";d3
500 d4=INT(((((bt*1.2)*gp))*100)+.5)/100
510 PRINT "tank LPG       F ";d4
520 PRINT "                ========== -"
530 PRINT "per tank       F ";(d3-d4)
540 PRINT
550 PRINT "LPG kan voor u p/mnd voordeliger
    zijn"
560 hu=((ap/at)+(wg/12))/((bt*bp)-
    ((bt*1.2)*gp))
570 PRINT
580 PRINT "U moet nu
    dan";INT(((hu*bt)*100)+.5)/100;"p/mndtanke
    n"
590 PRINT
600 PRINT "Dit is voor
    F";INT((((hu*bt)*bp)*100)+.5)/100;"aanbenz
    ine"
```

AMIMEMO

Het gebeurt niet vaak dat we hier een programma afdrukken waarvan het idee zo simpel is terwijl er toch zo weinigen opkomen om het te programmeren. Het programma was voor ons de uitkomst. Waar gaat het nu om, zult u zich afvragen. We tillen een tipje van de sluier op. Het gaat hier om de computertonische (niet Nederlands, we weten het!) vervanging van de aloude bekende agenda.



Hoe werkt het

Het programma zal, zoals van een agenda verwacht mag worden, data (het woord data van datum in dit geval) op het scherm afbeelden. Bij deze data zal natuurlijk ook een mededeling afgebeeld moeten worden. Nu een opmerking. Het is niet een specifiek Amiga programma daar hier de Gadgets en andere Amiga-specifieke zaken ontbreken. Het is zelfs zo dat met een paar kleine aanpassingen het programma op elke computer draaien kan. Dit met dien verstande dat men op deze computer over een C-compiler beschikt.

Data invoer

De data invoer gebeurt vanaf de command-line. U dient een kleine file te creeren waarin u vervolgens de data in dient te zetten. Dit kan met een editor zoals Ed gedaan worden of gewoon met het Copy commando zoals in het volgende voorbeeld gegeven is.

```
copy * datumfile
24-10-90 n Nieuwe 5
    1/4 flops halen.
...
```

(CTRL-\)

Zoals u ziet dient de uitvoer met CTRL-\ te worden beeindigd. In het bovenstaande voorbeeld is al enigzins naar voren gekomen hoe de data ingevoerd moet worden. De data invoer is opgedeeld in drie velden. Het eerste veld dient de datum te bevatten. Dit mag op meerdere manieren gebeuren. Voorbeelden zijn, 24-Okt-90, 24- Oktober-90, 24-10-1990 en andere. Bij een niet geldige datum zal het programma reageren met een foutmelding waarop u uw file dient te corrigeren. Het tweede veld kan maar twee waarden bevatten, namelijk 0 of 1 (of ook 'n' of 'j'). Hiermee geeft u aan of het veld repeterend is. De gegevens mogen dus niet uit de lijst verwijderd worden als de datum geweest is. Dit gebeurt wel als hier nee (=0 of 'n') vermeld staat. Als de datum repeterend moet zijn, dan zal het het volgende jaar weer opnieuw worden afgebeeld. Het laatste veld kan een willekeurige tekst bevatten. Toch is hier ook een beperking aan verbonden. De tekst mag namelijk niet langer zijn dan 80 tekens.

Programma gebruik

Heeft u eenmaal een datumfile opgesteld dan geeft u dit door aan het programma welke vervolgens uw file 'compileert' naar een voor het programma interpreteerbare file. De syntax luidt, Amimemo [datumfile]
De datumfile vermelding wordt tussen vierkante haken vermeld omdat u niet perse een datumfile op hoeft te geven. Als u dit namelijk niet doet zal het programma zoeken naar de file 'calfile' waarin de voor het programma relevante gegevens staan. Het programma leest dit uit en zal de voor u belangrijke data op het scherm vermelden.
Dus wilt u data aanmaken of toevoegen dan geeft u de naam van de file aan het programma Amimemo door. Geeft u geen naam door dan zalhet programma zoeken naar de gecompilede file waarin de belangrijke data opgeslagen zijn. Dezen worden vervolgens afgebeeld.

Instructies

Het programma is zowel met Aztek v3.6 en hoger en Lattice 5.0 compiler te compilen en te linken.

Aztek...

```
--------
cc AmiMemo.c +L
ln AmiMemo.o -lc32
```

Lattice...

```
----------
lc -L AmiMemo.c
```

Tot slot

Het programma is een op zich zelf staand programma, echter wie schenkt er genoegen in iedere keer het programma aan te moeten roepen om te weten te komen wat er op de agenda staat. Neem daarom maar van ons aan dat de beste manier om het programma aan te roepen is door dit vanuit de startup-sequence te laten gebeuren. Zet daarom onder in uw startup-sequence, AmiMemo. De rest zal dan automatisch gebeuren. Een voorbeeld,

```
setclock opt load
path df0:s add
cd s:
AmiMemo
```

Hierbij dient opgemerkt te worden dat in dit voorbeeld, zowel AmiMemo als de file 'calfile' in de s-directory aanwezig dienen te zijn. Heeft u op deze materie nog leuke opmerkingen of aanpassingen, laat het ons dan weten!

Johan & johan.



Het programma.

================

```
#include "stdio.h"
#ifndef LATTICE
 #include "functions.h"
#endif
#include "exec/types.h"
#include "exec/memory.h"
#include "libraries/dos.h"
#define STRING1 "Vergeet het niet, "
#define STRING2 "over "
#define STRING3 "VANDAAG"
#define STRING4 "week"
#define STRING5 "dagen"
#define STRING6 "dag"
#define STRING7 "morgen"
#define SPACE 0x20
char MonthName[13][10]={ "",
"JANuari","FEBruari","MaaRT","APRil","MEI","JUNi
"JULi","AUGustus","SEPtember","OKTober","NOVember"
    ,"DECember" };
int MonthLength[]={0,31,28,31,30,31,30,31,31,
    30,31,30,31 };
struct CalStruct {
 LONG Days;
 BOOL YrRpt;
 char *Message;
 struct CalStruct *NextEntry;
};
struct MsgList {
 struct CalStruct *CalEntry;
 struct MsgList *NextMsgList;
};
ULONG DateToDays(f_Buffer)
char *f_Buffer;
{
 int Day,Month,Year;
 BOOL Stop;
 ULONG Time;
 char *Temp,MntAbv[10];
 register int i,j,k,l;
 if(!(Temp=(char *)AllocMem(25,MEMF_CLEAR)))
  {
   printf("Geen mem. voor Temp\n");
   exit(20);
  }
 i=j=k=l=0;
 while((*(Temp+j++)=*(f_Buffer+i++))!='-' &&
    i<25);
  if(i == 25)
  {
   printf("Error in date-format\n");
   printf("dd-mm(or Monthname+abv.)-yy(yy)\n");
   printf(" ^\n");
   exit(20);
  }
 *(Temp+j-1)=0x00;
 Day=atoi(Temp);
 j=0;
 while((*(Temp+j++)=*(f_Buffer+i++))!='-' &&
    i<25);
 if(i == 25)
  {
   printf("Error in date-format\n");
   printf("dd-mm(or Monthname+abv.)-yy(yy)\n");
   printf("    ^\n");
   exit(20);
  }
 *(Temp+j-1)=0x00;
 Stop=FALSE;
 if(!(atoi(Temp)))
   {
    if(strlen(Temp) == 3)
     {
      for(j=1; j<13 && !Stop; j++)
       {
        for(k=0,l=0; k<strlen(Temp); k++)
         if(MonthName[j][k] > 64 &&
  MonthName[j][k] < 91)
           MntAbv[l++]=tolower(MonthName[j][k]);
        MntAbv[l]=NULL;
        MntAbv[0]=toupper(MntAbv[0]);
        if(!(strcmp(MntAbv,toupper(Temp))))
         {
          Month=j;
          Stop=TRUE;
         }
       }
     }
    else
     {
      Stop=FALSE;
      for(j=1; j<13 && !Stop; j++)
       {
        MntAbv[0]=MonthName[j][0];

        for(k=1; k<strlen(MonthName[j]); k++)
          MntAbv[k]=tolower(MonthName[j][k]);
        MntAbv[k]=NULL;
        if(!(strcmp(MntAbv,Temp)))
         {
          Month=j;
          Stop=TRUE;
         }
       }
     }
   }
  else
   Month=atoi(Temp);
  j=Time=0;
  while((*(Temp+j++)=*(f_Buffer+i++))!=SPACE &&
     i<25);
  if(i == 25)
   {
    printf("Error in date-format\n");
    printf("dd-mm(or Monthname+abv.)-yy(yy)\n");
    printf("                          ^\n");
    exit(20);
   }
  *(Temp+j-1)=NULL;
  Year=atoi(Temp);
  if(Year > 99)
    Year-=1900;
  Time=(Year-78)*365 + (int)((Year-78)/4);
  for(i=1; i<Month; i++)
    Time+=MonthLength[i];
  Time+=Day;
  FreeMem(Temp,25);
  return(Time-1);
}
void WriteLong(f_Val,fp)
FILE *fp;
ULONG f_Val;
{
 UBYTE a,b,c,d;
 a=f_Val&0xff;
 b=(f_Val>>8)&0xff;
 c=(f_Val>>16)&0xff;
 d=(f_Val>>24)&0xff;
 putc(d,fp);
 putc(c,fp);
 putc(b,fp);
 putc(a,fp);
```




```
}
ULONG ReadLong(fp)
FILE *fp;
{
 UBYTE byte1,byte2,byte3,byte4;
 ULONG len;
 char ch;
 byte1=getc(fp);
 byte2=getc(fp);
 byte3=getc(fp);
 byte4=getc(fp);
 if(!feof(fp))
  {
   if(!byte1 && !byte2)
     len=(ULONG)(byte1<<24 | byte2<<16 |
    byte3<<8 | byte4);
   else
    {
     printf("Format error in calfile, first TWO
    zero's missing!\n");
     while((ch=getc(fp))!=0x0a)
       printf("%c",ch);
     exit(20);
    }
  }
 else
   len=NULL;
 return(len);
}
struct CalStruct *FillCalList(fp)
FILE *fp;
{
 struct CalStruct *NewCalStruct;
 ULONG ReadTime;
 register int i;
 char DBuffer[80];
 if(!(ReadTime=ReadLong(fp)))
   return(NULL);
 else
  {
   if(!(NewCalStruct=(struct CalStruct *)
     AllocMem(sizeof(struct
    CalStruct),MEMF_CLEAR)))
      {
       printf("No mem. for new item res.\n");
       exit(20);
      }
   NewCalStruct->Days=ReadTime;
   NewCalStruct->YrRpt=(BOOL)getc(fp);
   i=0;
   while((DBuffer[i++]=getc(fp))!=0x0a && i<81);
   DBuffer[i]=NULL;
   if(i==81)
    {
     printf("Sourcefile error. Message exceeded
    80 characters.\n");
     exit(20);
    }
   if(!(NewCalStruct->Message=(char *)
     AllocMem(strlen(DBuffer),MEMF_CLEAR)))
      {
       printf("No mem. for Message\n");
       exit(20);
      }
   for(i=0; i<strlen(DBuffer)+1; i++)
     *(NewCalStruct->Message+i)=DBuffer[i];
   NewCalStruct->NextEntry=FillCalList(fp);
   return(NewCalStruct);
  }
}
void UpDateCalFile(Head,CurrDays)
struct CalStruct *Head;
ULONG CurrDays;
{
 FILE *fp,*fopen();
 struct CalStruct *Item;
 register int i;
 unlink("calfile");
 if(!(fp=fopen("calfile","w")))
  {
   printf("Update file not open after
    destruction.\nAll data is lost!\n");
   printf("Please create new calfile\n");
   exit(20);
  }
 Item=Head;
 do{
   if(Item->Days > CurrDays)
    {
     WriteLong(Item->Days,fp);
     putc(Item->YrRpt,fp);

     for(i=0; i<strlen(Item->Message); i++)
       putc(*(Item->Message+i),fp);
    }
   else
    {
     if(Item->YrRpt)
      {

    WriteLong(Item->Days+365+((((Item->Days+365)/3
    65)%4)==0),fp);
       putc(Item->YrRpt,fp);
       for(i=0; i<strlen(Item->Message); i++)
         putc(*(Item->Message+i),fp);
      }
    }
 }while(Item=Item->NextEntry);

 fclose(fp);
}
struct CalStruct *ReadCalFile()
{
 FILE *fp,*fopen();
 struct CalStruct *Head;
 if(!(fp=fopen("calfile","r")))
  {
   printf("File not exists\n");
   exit(20);
  }
 Head=FillCalList(fp);
 fclose(fp);
 return(Head);
}
struct MsgList *SearchForEntry(f_Item,f_SrchDay)
struct CalStruct *f_Item;
ULONG f_SrchDay;
{
 struct MsgList *SrchList;
 if(!f_Item)
   return(NULL);
 else
  {
   if(f_Item->Days == f_SrchDay)
    {
     if(!(SrchList=(struct MsgList *)
       AllocMem(sizeof(struct
    MsgList),MEMF_CLEAR)))
         {
          printf("No mem. for SrchList entry\n");
     exit(20);
    }
      SrchList->CalEntry=f_Item;
```



```
    SrchList->NextMsgList=SearchForEntry(f_Item->N
    extEntry,f_SrchDay);
      return(SrchList);
     }
    else

    return(SearchForEntry(f_Item->NextEntry,f_Srch
    Day));
  }
}
void FreeSrchList(f_SrchList)
struct MsgList *f_SrchList;
{
 struct MsgList *TempMsg,*Item;
 Item=f_SrchList;
 if(Item)
  {
   do{
    TempMsg=Item->NextMsgList;
    FreeMem(Item,sizeof(struct MsgList));
    Item=TempMsg;
   }while(Item);
  }
}
void FreeCalList(f_DateList)
struct CalStruct *f_DateList;
{
 struct CalStruct *TempCal,*Item;
 Item=f_DateList;
 if(Item)
  {
   do{
    TempCal=Item->NextEntry;

    FreeMem(Item->Message,strlen(Item->Message));
    FreeMem(Item,sizeof(struct CalStruct));
    Item=TempCal;
   }while(Item);
  }
}
main(argc,argv)
int argc;
char *argv[];
{
 struct CalStruct *Head;
 struct MsgList *Entries,*MsgListHead;
 struct DateStamp ds;
 SHORT DaysAhead;
 ULONG Time;
 char ch;
 FILE *ReadFp,*WriteFp;
 BOOL Rpt;
 register int i;
 char DBuffer[80];
 DateStamp(&ds);
 if(!(strlen(argv[1])))
  {
   Head=ReadCalFile();
   UpDateCalFile(Head,ds.ds_Days);
   DaysAhead=7;
   while(DaysAhead != -1)
    {

    if((Entries=MsgListHead=SearchForEntry(Head,ds
    .ds_Days+DaysAhead)))
      {
       do
        {
         switch(DaysAhead)
          {
           case 7:
             sprintf(DBuffer,"%s%s1 %s:
%s",STRING1,STRING2,STRING4,
           Entries->CalEntry->Message);
             break;
           case 1:
             sprintf(DBuffer,"%s%s:
%s",STRING1,STRING7,
                  Entries->CalEntry->Message);
             break;
           case 0:
             sprintf(DBuffer,"%s%s:
%s",STRING1,STRING3,
          Entries->CalEntry->Message);
             break;
           default:
             sprintf(DBuffer,"%s%s%d %s:
%s",STRING1,STRING2,DaysAhead,STRING5,
                  Entries->CalEntry->Message);
             break;
          }
         printf("%s",DBuffer);
        }while(Entries=Entries->NextMsgList);
       FreeSrchList(MsgListHead);
      }
     DaysAhead--;
    }
   FreeCalList(Head);
  }
 else
  {
   if(!(ReadFp=fopen(argv[1],"r")))
    {
     printf("Error in read file open\n");
     exit(20);
    }
   if(!(WriteFp=fopen("calfile","r")))
    WriteFp=fopen("calfile","w");
   else
    {
     fclose(WriteFp);
     WriteFp=fopen("calfile","a");
    }
   ch=getc(ReadFp);
   do{
    i=0;
    while(ch!=SPACE)
     {
      DBuffer[i++]=ch;
      ch=getc(ReadFp);
     }
    DBuffer[i]=ch;
    Time=DateToDays(DBuffer);
    ch=getc(ReadFp);
    Rpt=(strcmp('y',ch)==TRUE ||
         strcmp('j',ch)==TRUE ||
     strcmp('1',ch)==TRUE);
    if(Time < ds.ds_Days && Rpt)
     Time+=365+((((Time+365)/365)%4)==0);
    ch=getc(ReadFp); /* skip space */
    if(Time >= ds.ds_Days)
     {
      WriteLong(Time,WriteFp);
      putc(Rpt,WriteFp);
     }
    do{
      ch=getc(ReadFp);
      if(Time >= ds.ds_Days)
        putc(ch,WriteFp);
     }while(ch!=0x0a);
   } while((ch=getc(ReadFp))!=EOF);
  fclose(WriteFp);
  fclose(ReadFp);
```